Small, but important: Traffic light proposals for detecting small traffic lights and beyond

Tom Sanitz[1,2,*], Christian Wilms[2,*], and Simone Frintrop[2]

[1] Ibeo Automotive Systems GmbH, Hamburg, Germany
[2] Computer Vision Group, University of Hamburg, Hamburg, Germany
{christian.wilms,simone.frintrop}@uni-hamburg.de

∗ indicates equal contribution

Abstract. Traffic light detection is a challenging problem in the context of self-driving cars and driver assistance systems. While most existing systems produce good results on large traffic lights, detecting small and tiny ones is often overlooked. A key problem here is the inherent downsampling in CNNs, leading to low-resolution features for detection. To mitigate this problem, we propose a new traffic light detection system, comprising a novel traffic light proposal generator that utilizes findings from general object proposal generation, fine-grained multi-scale features, and attention for efficient processing. Moreover, we design a new detection head for classifying and refining our proposals. We evaluate our system on three challenging, publicly available datasets and compare it against six methods. The results show substantial improvements of at least 12.6% on small and tiny traffic lights, as well as strong results across all sizes of traffic lights.

1 Introduction

Traffic light detection, which involves locating traffic lights and classifying their state, is an essential task for self-driving cars and driver assistance systems. Due to the complex nature of urban environments with several intersections, heavy traffic, and distracting objects, detecting traffic lights is challenging [14]. Moreover, safe driving in such environments is challenging to humans as well, leading to stress, oversights, and potentially fatal accidents [12]. Hence, support by driver assistance systems is of great importance in such environments.

Several approaches for traffic light detection were proposed, mostly based on standard object detectors [3, 1, 23, 20]. However, the detection of traffic lights appearing small or tiny on the image plane remains a problem. A major reason is the inherent subsampling in CNNs to extract semantically rich features. Moreover, small and tiny traffic lights are much smaller than objects typically annotated in object detection datasets like COCO [17]. Hence, standard object detectors might not be well-suited for the problem of detecting such traffic lights, as visible in the results in Fig. 1. Despite the difficulties, small and tiny traffic

Image TL-SSD [20] Ours

Fig. 1: Qualitative results of TL-SSD [20] and our proposed traffic light detection system, which highlight the weak performance of current systems on small and tiny traffic lights. The color indicates the assigned traffic light state. Dark blue denotes the state *off*, while white denotes missed or misclassified traffic lights. Arrows highlight traffic lights for better visibility.

lights are essential for safe, efficient, and eco-friendly driving. For instance, detecting traffic lights early, when they are still far away and small or tiny on the image plane, allows a car to slowly approach a traffic light showing a stop signal. This reduces noise emission, fuel consumption, and carbon emission [33, 26].

Several directions were proposed to improve the detection of small and tiny objects in computer vision literature [4]. Prominent directions include tiling approaches [2, 22, 31, 32], which process sub-images to increase the objects' relative size. However, this leads to substantially increased runtimes [32]. Another direction is a multi-scale feature representation, combing semantically rich, low-resolution features with high-resolution features lacking high-level semantics [6, 16]. A third direction are object proposal generators to discover small objects [10, 29, 19]. Since the latter two directions do not suffer from substantially increased runtime, we follow them to improve the detection of small and tiny traffic lights.

In this paper, we propose a new traffic light detection system, focusing on small and tiny traffic lights, while also improving the detection across all sizes of traffic lights. First, to better localize small and tiny traffic lights, we propose a new traffic light proposal generator. It utilizes the one-shot paradigm from object proposal generation [9], introduces an improved multi-scale feature representation to create semantically rich features at high resolution, and employs attention for increased efficiency. To classify and refine the proposals, we adapt the Faster R-CNN framework [25] utilizing our high-quality proposals and a new traffic light detection head. Our extensive evaluation on three challenging, publicly available datasets demonstrates the effectiveness of our approach for detecting small and tiny traffic lights, as well as traffic lights of all sizes. Across all three datasets, we outperform all other tested methods and strong baselines.

Overall, our contributions are threefold:

- A novel traffic light detection system focusing on small and tiny traffic lights.
- A new traffic light proposal generator using a multi-scale feature representation and attention.
- An extensive evaluation on three challenging datasets, outperforming all other tested methods and strong baselines across all datasets.

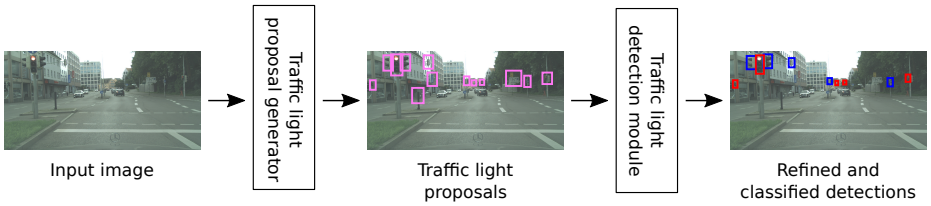

Fig. 2: Overview of our traffic light detection system. First, our novel traffic light proposal generator creates traffic light proposals (pink boxes) covering possible traffic light locations. Subsequently, our detection module refines and classifies each proposal (red/blue boxes).

2 Related Work

Traffic light detection has attracted considerable attention over the last decade. While early approaches used traditional techniques including color thresholding or extracting shape information, see [12] for a survey, CNNs have been utilized recently. Subsequently, we review the most important and relevant approaches utilizing RGB data only.

Most recent CNN-based traffic light detection approaches extend standard object detectors like SSD [18], YOLO [24], or Faster R-CNN [25]. For instance, [20] and [14] improve SSD's detection quality for small traffic lights by proposing modified anchors or an additional classifier. Various versions of YOLO are also adapted for traffic light detection [3, 15, 27]. To address the challenge of tiny traffic lights, [3] use a tiling approach, while [27] combine features from various layers of the network for detection. Finally, Faster R-CNN is commonly utilized for traffic light detection with adjustments to the anchor boxes [1], backbones [1, 13], color spaces [13], and the classification head [1, 23, 7].

Apart from approaches based on standard object detectors, few notable approaches exist. [21] use a fully convolutional approach with a hierarchical per-pixel classifier. [28] propose a heuristics-based proposal generator to improve the detection of small traffic lights. Recently, [34] introduced a detection method for small objects that disentangles the features of different classes and instances, which is also applied to traffic light detection.

Overall, for the challenging detection of small and tiny traffic lights, most approaches use variations of standard object detectors or utilize heuristics-based approaches. In contrast, we propose a new traffic light detection system with a dedicated traffic light proposal generator that introduces an improved multi-scale feature representation to create semantically rich features at high resolution as well as attention for increased efficiency.

3 Method

This section introduces our new traffic light detection system, visualized in Fig. 2. Given an input image, our novel traffic light proposal generator, described in

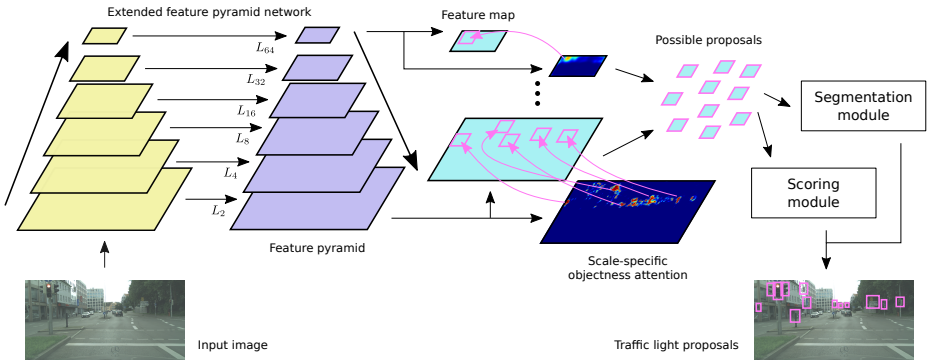

Fig. 3: Detailed view of our traffic light proposal generator. Our extended feature pyramid network (yellow and violet pyramids) extracts semantically rich features across all levels/resolutions (L_2 to L_{64}), yielding a multi-scale feature pyramid (violet pyramid). For each level, we generate a feature map (large cyan boxes) and a scale-specific objectness attention map highlighting possible traffic light locations. Subsequently, we extract windows (possible proposals) at high-attention locations across the feature pyramid. Finally, a segmentation and a score are generated per window to create a ranked list of traffic light proposals.

Sec. 3.1, extracts possible traffic light locations. The proposal generator explicitly addresses the challenging localization of small traffic lights by improving the resolution of the multi-scale feature representation and utilizes attention for efficient processing. Subsequently, our traffic light detection module refines the proposals and assigns them to a traffic light state based on a new traffic light detection head (see Sec. 3.2). Finally, we discuss the training of our system in Sec 3.3.

3.1 Traffic Light Proposal Generator

The first stage of our traffic light detection system consists of our novel traffic light proposal generator, visualized in Fig. 3. It is designed to locate all traffic lights and reduce the search space for the subsequent traffic light detection module. Similar to recent object proposal generators [9, 29, 30], we follow the efficient one-shot approach [9]. Hence, the input image is processed only once by the backbone network, yielding a feature representation that is further subsampled to create a multi-scale feature pyramid. By extracting fixed-size windows from the feature pyramid, proposals for objects of different sizes are generated.

However, simply following this approach is insufficient for traffic light detection with many tiny objects. The additional subsampling to create a feature pyramid on top of the backbone leads to low-resolution feature maps and missing tiny objects. To circumvent this problem, we use the backbone itself as a multi-scale feature pyramid, introducing less subsampling. Yet, two issues arise.

First, the features from early layers of backbones like ResNets [8] are not semantically rich. To generate a feature pyramid that is semantically rich across all levels, we employ and extend the Feature Pyramid Network (FPN) [16] as backbone (yellow and violet pyramids in Fig. 3). The FPN combines a typical CNN-backbone with top-down and lateral connections to create a semantically rich representation across all levels (resolutions) of the feature pyramid (violet pyramid in Fig. 3). In Fig. 3, the levels are denoted by L_n with the downsampling factor n. We further add a new level to the feature pyramid for tiny objects (L_2 in Fig. 3). This multi-scale feature pyramid serves as the base for our proposal generator.

The second issue is the large number of possible locations for extracting windows, i.e., possible traffic light proposals, in our feature pyramid due to the high resolution. To address this issue, we utilize scale-specific objectness attention proposed by [29]. The attention maps are learned per feature pyramid level (scale-specific) and focus the window extraction as well as further processing on the most relevant areas, omitting the background.

Based on the attention maps, we extract all relevant, i.e., high attention, windows of size 10×10 across all levels of our feature pyramid (small cyan boxes in Fig. 3). Subsequently, we use the common head structure of [9] to score the possible proposals and generate a segmentation per proposal. The final result of the traffic light proposal generator are n traffic light proposals with bounding box coordinates and a score to process only the most relevant ones.

3.2 Traffic Light Detection Module

To assign the proposals generated by our traffic light proposal generator to the traffic light states and refine their locations, we apply our traffic light detection module, which is inspired by the Faster R-CNN architecture [25]. First, we extract a feature representation of the image utilizing an FPN backbone. Note that we can share this backbone with the proposal generator.

Given the feature representation of the image and our proposals, we apply region of interest pooling per proposal, leading to a 7×7 feature map per proposal. Subsequently, each proposal is processed by our new traffic light detection head consisting of four fully-connected layers with 2048 neurons each. Based on these features, the detection head refines the traffic light location and assigns a traffic light state or the background label to each proposal.

3.3 Training

This section describes the training of our traffic light detection system. First, we train our traffic light proposal generator utilizing the annotated data of the respective dataset, omitting any state information. We follow [9, 29] for defining the respective loss functions of the attention modules and the head for segmenting as well as scoring the traffic light proposals.

After training the traffic light proposal generator, we utilize it to generate positive and negative training samples for our traffic light detection module. A proposal is regarded as a positive sample if it has an Intersection over Union (IoU) of at least 0.5 with any annotated traffic light, while proposals with an IoU below 0.3 are regarded as negative samples. The rest of the training regime is similar to [25].

4 Experiments

We evaluate our approach on three challenging, publicly available datasets and compare it to six systems across the different datasets. The datasets are the Bosch Small Traffic Lights Dataset (BSTLD) [3], the DriveU Traffic Light Dataset (DTLD) [5], and the recently published dataset Cityscapes TL++ (CS-TL) [11]. They consist of 1978 to 40953 images with an average of 2.4 to 6.6 annotated traffic lights per image. All images show traffic scenes captured from the perspective of the driver and are of high quality. The number of traffic light states differs between the datasets. While all datasets include the states *stop*, *warning*, and *go*, DTLD has *stop/warning* annotated and all datasets include the states *off* or *unknown*. Note that we adapted all systems to the respective number of traffic light states contained in a dataset.

We compare our approach to different methods per dataset due to the limited availability of results and code as well as varying dataset splits. On BSTLD, we compare to [3], [23], and the SSD and Faster R-CNN baselines[3] provided by [3]. For DTLD, we generate the results of [20] with their publicly available system[4]. Since no results are publicly available yet on the CS-TL dataset, we compare to a strong baseline using Faster R-CNN [25] with an FPN backbone [16].

To assess the quality of traffic light detection results, we use the mean Average Precision (mAP) with an IoU of 0.5 as the main measure, following [1, 23, 15, 34]. For a more detailed analysis, we report state-specific and size-specific results. While the traffic light states are fixed by the datasets, we define four relative size ranges, based on the ratio of an annotated traffic light's area and the image area. The ranges for relative size a are tiny ($a \leq 0.01\%$), small ($0.01\% < a \leq 0.03\%$), medium ($0.03\% < a \leq 0.05\%$), and large ($0.05\% < a$), denoted as mAP_T, mAP_S, mAP_M, and mAP_L. The ranges are determined such that across all datasets, each class comprises between 20% and 30% of the annotated traffic lights. Finally, we also report $\text{mAP}_{weighted}$ on BSTLD, defined by [3], which incorporates the distribution of traffic light states in the dataset.

The subsequent sections discuss the quantitative results, selected qualitative results, and two ablation studies justifying design choices.

4.1 Quantitative Results

The quantitative results on BSTLD (see Tab. 1), DTLD (see Tab. 2), and the CS-TL dataset (see Tab. 3) all show similar trends. Across all datasets, our

[3] Results taken from https://github.com/bosch-ros-pkg/bstld
[4] https://github.com/julimueller/tl_ssd

Table 1: Traffic light detection results on the BSTLD test set in terms of mAP, $\text{mAP}_{weighted}$, and state-specific mAPs.

System	$\text{mAP}_{weighted}$	mAP	mAP_{stop}	$\text{mAP}_{warning}$	mAP_{go}	mAP_{off}
YOLO TLD [3]	0.360	-	-	-	-	-
HDA [23]	0.530	-	-	-	-	-
SSD TLD [3]	0.600	0.410	0.550	0.410	0.680	**0.000**
Faster R-CNN NAS [3]	0.650	0.430	0.660	0.330	0.710	**0.000**
Ours	**0.710**	**0.572**	**0.678**	**0.855**	**0.753**	**0.000**

Table 2: Traffic light detection results on the DTLD test set in terms of mAP and state-specific mAPs.

System	mAP	mAP_{stop}	$\text{mAP}_{stop/warning}$	$\text{mAP}_{warning}$	mAP_{go}	mAP_{off}
TL-SSD [20]	0.329	0.439	0.283	0.167	0.583	0.010
Ours	**0.552**	**0.699**	**0.564**	**0.564**	**0.789**	**0.142**

proposed system outperforms all other methods and baselines in terms of mAP. Within the datasets, the results for the individual traffic light states vary. One reason is the imbalance of annotations in the datasets. For instance, on DTLD, the states *stop* and *go* amount to almost 85% of the annotations in the dataset. Hence, the results for those states are substantially better than for the other states. Another reason is the intra-class diversity of the states *off* and *unknown*.

On BSTLD, the results in Tab. 1 show that our proposed traffic light detection system outperforms both SSD TLD and Faster R-CNN NAS [3] based on standard object detectors by 39.5% and 33.0% in terms of mAP. For the state *warning*, the improvement is even up to 159%. Similarly, in terms of $\text{mAP}_{weighted}$, our system outperforms all other methods by up to 97.2%.

The results on DTLD in Tab. 2 comparing TL-SSD [20] based on a standard object detector to our proposed system show again a strong improvement across all traffic light states (+67.8%). On the recently published CS-TL dataset, we compare to the strong Faster R-CNN+FPN [25, 16] baseline (see Tab. 3). While across all states, i.e. overall, our traffic light detection system outperforms the baseline (+1%), the per-state results favor both systems twice.

Table 3: Traffic light detection results on the CS-TL test set in terms of mAP and state-specific mAPs.

System	mAP	mAP_{stop}	$\text{mAP}_{warning}$	mAP_{go}	$\text{mAP}_{unknown}$
Faster R-CNN+FPN [25, 16]	0.496	**0.645**	**0.381**	0.626	0.332
Ours	**0.500**	0.638	0.338	**0.650**	**0.375**

Table 4: Size-specific traffic light detection results on the test splits of BSTLD, DTLD, and the CS-TL dataset. T, S, M, and L denote traffic light size ranges tiny, small, medium, and large.

Dataset	System	mAP_T	mAP_S	mAP_M	mAP_L
BSTLD	Ours	0.061	0.508	0.518	0.679
DTLD	TL-SSD [20]	0.079	0.299	0.331	0.530
	Ours	**0.286**	**0.625**	**0.672**	**0.674**
CS-TL	Faster R-CNN+FPN [25, 16]	0.207	0.387	0.385	**0.608**
	Ours	**0.233**	**0.421**	**0.400**	0.531

Image TL-SSD [20] Ours

Fig. 4: Qualitative results of TL-SSD [20] and our proposed traffic light detection system on the DTLD test set. The color indicates the assigned traffic light state. Blue denotes the state *off*, while white denotes missed or misclassified traffic lights. Arrows highlight traffic lights for better visibility.

Analyzing the detection performance in more detail, Tab. 4 shows the results for different relative sizes of traffic lights. Across all systems and datasets, the results on tiny traffic lights are substantially worse compared to the three other size ranges, with a drop of up to 91%. Note that the traffic light in BSTLD are smaller, resulting in worse results compared to the other datasets. Due to the lack of publicly available results and the novel and custom nature of the evaluation, we can only compare to TL-SSD on DTLD and the Faster R-CNN+FPN baseline on the CS-TL dataset. On both datasets, we outperform the other methods on tiny traffic lights in terms of mAP_T (+262% and +12.6%). This confirms the strong performance of our system on such traffic lights. Across the other size ranges, we also outperform TL-SSD on DTLD by an average of 79.7%. On the CS-TL dataset, we outperform the strong Faster R-CNN+FPN baseline on small and medium traffic lights, with a slight drop on larger ones.

Overall, the quantitative results across three datasets show the strong performance of our traffic light detection system. Particularly, the size-specific results indicate the substantially improved detection of small and tiny traffic lights.

Fig. 5: Qualitative results of the Faster R-CNN+FPN [25, 16] baseline and our proposed traffic light detection system on the CS-TL test set. The color indicates the assigned traffic light state. Blue denotes the state *unknown*, while white denotes missed or misclassified traffic lights. Arrows highlight traffic lights for better visibility.

4.2 Qualitative Results

The qualitative results in Fig. 4 on DTLD and in Fig. 5 on the CS-TL dataset support the quantitative results. Comparing the results of TL-SSD and our system on DTLD in Fig. 4, it is clearly visible that both systems detect most larger traffic lights. However, the small and tiny traffic lights are only regularly detected by our system. TL-SSD is only able to detect one smaller traffic light, while our system detects almost all of them in the presented examples.

Switching to the recently published CS-TL dataset, Fig. 5 shows a comparison of our results and the strong Faster R-CNN+FPN baseline. The results across the four images again show a strong performance of both systems on larger traffic lights, with few exceptions like in the complex example in the final row. Investigating the results in the first two rows in more detail, it is visible that despite a rather low scene complexity, only our system is able to consistently detect smaller traffic lights like the two traffic lights with unknown state (blue) in the second row. The lower two rows show more complex scenes with several traffic lights. Despite the complexity, our system again detects most traffic lights and

Table 5: Traffic light proposal generation results for three proposal generators on BSTLD in terms of AR for 1000 and 5000 proposals.

Proposal Generator	AR@1000	AR@5000
$\text{AttentionMask}_8^{128}$ [29]	0.180	0.180
$\text{AttentionMask}_4^{96}$ [29, 32]	**0.390**	0.400
Ours	**0.390**	**0.450**

Table 6: Traffic light detection results for three detection heads on BSTLD in terms of mAP.

Head Structure	mAP
2×1024 [25]	0.554
4×2048 (ours)	**0.563**
5×2048	0.546

outperforms Faster R-CNN+FPN, especially on smaller instances. Still, some traffic lights are missed by our system due to low contrast (see last row).

Overall, these results correspond well to the findings in Tab. 4. Our system shows a stronger performance on small and tiny traffic lights compared to other systems, while generally accomplishing very good overall results.

4.3 Ablation Studies

This section investigates the choice of the traffic light proposal generator and the detection head. Both ablation studies were conducted on BSTLD.

Proposal Generator In Sec. 3.1, we proposed a new traffic light proposal generator for locating possible traffic lights. To show the benefit of our proposal generator, which is explicitly designed to discover small and tiny traffic lights, we compare it to two variations of AttentionMask [29]. AttentionMask is a general-purpose object proposal generator designed to discover small objects. The differences between the variations and our system are the spatial resolution of the most fine-grained feature map in the feature pyramid and the usage of an FPN-based backbone (see Sec. 3.1). While the two versions of AttentionMask utilize a feature map with a downscale factor of 8 ($\text{AttentionMask}_8^{128}$ [29]) and 4 ($\text{AttentionMask}_4^{96}$ [32]) as the base of the feature pyramid, we use a feature map with downscale factor 2 as a result of the FPN-based backbone. Note that $\text{AttentionMask}_8^{128}$ and $\text{AttentionMask}_4^{96}$ do not use an FPN-based backbone. The results in Tab. 5 on BSTLD in terms of Average Recall[5] (AR) for 1000 and 5000 proposals show that we strongly outperform original AttentionMask with an improvement of 150% on AR@5000. Compared to the variation $\text{AttentionMask}_4^{96}$, the improvement is still 12.5%. Therefore, our traffic light proposal generator based on the extended FPN-based feature pyramid outperforms general-purpose object proposal generators on the traffic light localization task.

Detection Head The detection head in our traffic light detection module refines the traffic light proposals and assigns them a traffic light state. As discussed

[5] Average Recall assesses how many annotated traffic lights are recalled and how precisely they are located, given a specified number of proposals.

in Sec. 3.2, our new detection head comprises four fully-connected layers with 2048 neurons each. We compare this design to the detection head of Faster R-CNN [25], which has two layers with 1024 neurons, and a larger detection head with five layers and 2048 neurons each. The results in terms of mAP on BSTLD in Tab. 6 indicate that our new detection head outperforms Faster R-CNN's detection head (+1.6%) as well as the larger detection head (+3.1%). Hence, our detection head is a good choice for the traffic light detection task.

5 Conclusion

In this paper, we addressed the problem of traffic light detection. We specifically focused on the challenging detection of small and tiny traffic lights, which are important for safe, efficient, and eco-friendly driving. Our approach consists of (i) a novel traffic light proposal generator combining the one-shot approach from object proposal generation with fine-grained multi-scale features as well as attention, and (ii) a detection module featuring a new traffic light detection head. The extensive evaluation across three datasets and six methods clearly shows the strong performance of our novel system on small and tiny traffic lights (at least +12.6%), as well as a strong overall performance on traffic lights of all sizes. Thus, our system can improve safe, efficient, and eco-friendly driving.

References

1. Bach, M., Stumper, D., Dietmayer, K.: Deep convolutional traffic light recognition for automated driving. In: ITSC (2018)
2. Bargoti, S., Underwood, J.: Deep fruit detection in orchards. In: ICRA (2017)
3. Behrendt, K., Novak, L., Botros, R.: A deep learning approach to traffic lights: Detection, tracking, and classification. In: ICRA (2017)
4. Chen, G., Wang, H., Chen, K., Li, Z., Song, Z., Liu, Y., Chen, W., Knoll, A.: A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. IEEE Trans. Syst. Man Cybern.: Syst. **52**(2) (2020)
5. Fregin, A., Muller, J., Krebel, U., Dietmayer, K.: The DriveU traffic light dataset: Introduction and comparison with existing datasets. In: ICRA (2018)
6. Gong, Y., Yu, X., Ding, Y., Peng, X., Zhao, J., Han, Z.: Effective fusion factor in FPN for tiny object detection. In: WACV (2021)
7. Gupta, A., Choudhary, A.: A framework for traffic light detection and recognition using deep learning and Grassmann manifolds. In: IV (2019)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
9. Hu, H., Lan, S., Jiang, Y., Cao, Z., Sha, F.: FastMask: Segment multi-scale object candidates in one shot. In: CVPR (2017)
10. Hu, P., Ramanan, D.: Finding tiny faces. In: CVPR (2017)
11. Janosovits, J.: Cityscapes TL++: Semantic traffic light annotations for the cityscapes dataset. In: ICRA (2022)
12. Jensen, M.B., Philipsen, M.P., Møgelmose, A., Moeslund, T.B., Trivedi, M.M.: Vision for looking at traffic lights: Issues, survey, and perspectives. T-ITS **17**(7) (2016)

13. Kim, H.K., Park, J.H., Jung, H.Y.: An efficient color space for deep-learning based traffic light recognition. JAT (2018)
14. Kim, J., Cho, H., Hwangbo, M., Choi, J., Canny, J., Kwon, Y.P.: Deep traffic light detection for self-driving cars from a large-scale dataset. In: ITSC (2018)
15. Lee, E., Kim, D.: Accurate traffic light detection using deep neural network with focal regression loss. IMAVIS **87** (2019)
16. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: ECCV (2016)
19. Lu, H.F., Du, X., Chang, P.L.: Toward scale-invariance and position-sensitive region proposal networks. In: ECCV (2018)
20. Müller, J., Dietmayer, K.: Detecting traffic lights by single shot detection. In: ITSC (2018)
21. Ouyang, Z., Niu, J., Liu, Y., Guizani, M.: Deep CNN-based real-time traffic light detector for self-driving vehicles. IEEE Trans. Mob. Comput. **19**(2) (2019)
22. Ozge Unel, F., Ozkalayci, B.O., Cigla, C.: The power of tiling for small object detection. In: CVPRW (2019)
23. Pon, A., Adrienko, O., Harakeh, A., Waslander, S.L.: A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. In: CRV (2018)
24. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
25. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks (2015)
26. Rittger, L., Schmidt, G., Maag, C., Kiesel, A.: Driving behaviour at traffic light intersections. Cogn. Technol. Work **17** (2015)
27. Wang, Q., Zhang, Q., Liang, X., Wang, Y., Zhou, C., Mikulovich, V.I.: Traffic lights detection and recognition method based on the improved YOLOv4 algorithm. Sensors **22**(1) (2022)
28. Weber, M., Huber, M., Zöllner, J.M.: HDTLR: A CNN based hierarchical detector for traffic lights. In: ITSC (2018)
29. Wilms, C., Frintrop, S.: AttentionMask: Attentive, efficient object proposal generation focusing on small objects. In: ACCV (2018)
30. Wilms, C., Frintrop, S.: Superpixel-based refinement for object proposal generation. In: ICPR (2021)
31. Wilms, C., Heid, R., Sadeghi, M.A., Ribbrock, A., Frintrop, S.: Which airline is this? Airline logo detection in real-world weather conditions. In: ICPR (2021)
32. Wilms, C., Johanson, R., Frintrop, S.: Localizing small apples in complex apple orchard environments. arXiv preprint arXiv:2202.11372 (2022)
33. Wu, C., Zhao, G., Ou, B.: A fuel economy optimization system with applications in vehicles with human drivers and autonomous vehicles. Transp. Res. D **16**(7) (2011)
34. Yang, X., Yan, J., Liao, W., Yang, X., Tang, J., He, T.: SCRDET++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. TPAMI **45**(2) (2022)