sensors

Article

Traffic Light Recognition Based on Binary Semantic Segmentation Network

Hyun-Koo Kim [1], Kook-Yeol Yoo [1], Ju H. Park [2] and Ho-Youl Jung [1,*]

[1] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, Korea; kim-hk@ynu.ac.kr (H.-K.K.); kyoo@ynu.ac.kr (K.-Y.Y.)
[2] Department of Electrical Engineering, Yeungnam University, Gyeongsan 38544, Korea; jessie@ynu.ac.kr
* Correspondence: hoyoul@yu.ac.kr; Tel.: +82-53-810-3545

Received: 29 January 2019; Accepted: 3 April 2019; Published: 10 April 2019

Abstract: A traffic light recognition system is a very important building block in an advanced driving assistance system and an autonomous vehicle system. In this paper, we propose a two-staged deep-learning-based traffic light recognition method that consists of a pixel-wise semantic segmentation technique and a novel fully convolutional network. For candidate detection, we employ a binary-semantic segmentation network that is suitable for detecting small objects such as traffic lights. Connected components labeling with an eight-connected neighborhood is applied to obtain bounding boxes of candidate regions, instead of the computationally demanding region proposal and regression processes of conventional methods. A fully convolutional network including a convolution layer with three filters of (1 × 1) at the beginning is designed and implemented for traffic light classification, as traffic lights have only a set number of colors. The simulation results show that the proposed traffic light recognition method outperforms the conventional two-staged object detection method in terms of recognition performance, and remarkably reduces the computational complexity and hardware requirements. This framework can be a useful network design guideline for the detection and recognition of small objects, including traffic lights.

Keywords: advanced driver assistance system; artificial neural networks; binary semantic segmentation; deep learning; traffic light detection; traffic light recognition

1. Introduction

A traffic light (TL) recognition system is a very important building block in an advanced driving assistance system (ADAS) and an autonomous vehicle system [1–3]. Information from various sensors, such as a stereo camera, radar sensor, digital-map, and GPS, are combined to improve recognition performance by predicting the position of the TL [4–12]. Though the sensor fusion methods give better performance, they suffer from high sensor costs, high computational complexity, and sophisticated manipulations such as sensor calibration and spatio-temporal synchronization among the various sensors [4–12]. For these reasons, research to improve the recognition performance using a single RGB camera has been continuously and thoroughly conducted in the literature [13–27]. Many methods have been developed based on signal processing and computer vision techniques, but they are still highly sensitive to environmental variations such as illumination change and noise.

Recently, deep-learning methods, such as region-convolutional neural network (R-CNN) [28], Fast R-CNN [29], Faster R-CNN [30], region-based fully convolutional networks (R-FCN) [31], Mask R-CNN [32], you only look once (YOLO) [33–35], and single shot multibox detector (SSD) [36,37] have showed innovative performance in the object recognition field. The first five methods are composed of two stages: region detection and object classification [28–32]. The other methods combine both tasks into a single stage [33–37]. The two-staged methods give better performance, with the sacrifice of increased computational complexity. On the other hand, the single staged methods can detect objects in real-time, but produce low detection performance.

The above-mentioned methods cannot produce sufficient performance for detecting small TLs for an ADAS application, because they are originally developed for detecting general objects such as vehicles, pedestrians, and animals [38–40]. For the well-known Bosch traffic dataset [26], 89% of TLs are classified as small objects [40]. It should be noted that an object with less than 32 × 32 pixels is defined as a small object in the COCO dataset [41].

The deep-learning approach has deep depths of multiple layers with convolutional filtering and follows pooling, i.e., down-sampling. The deep layers produce various excellent features by using a large receptive field, i.e., the whole input image. For the TL recognition case, the large receptive field should be adjusted, because the TL has very small spatial resolution. Considering the intrinsic performance limitation of the single-staged approaches, it would be important to reduce the computational complexity of the two-staged approaches for a TL detection application.

In this paper, we propose a two-staged deep-learning-based traffic light recognition method that consists of a pixel-wise semantic segmentation technique and a novel fully convolutional network (FCN). The proposed method works in a manageable computational complexity and with sufficient recognition performance. For the detection of small objects, pixel-wise semantic segmentation [42–45] is applied to detect the TL candidate regions. To remove the computationally-demanding candidate detection and regression operations in the conventional two-staged approach, a region segmentation method in computer vision is adopted for real-time processing. By contrast, in the case of classification of TL types, we notice two important facts: (1) the resolutions of candidate TL regions are variable, unlike that of conventional deep-learning-based classification with a fixed input resolution, and (2) the TLs have only a set number of colors, such as red, green, yellow, and black (TL back-plate). The pre-processing of R-CNN [28] for input resolution variation is adopted in the proposed classification, i.e., the input region is warped to have the required resolution. For the proper color space transformation, (1 × 1) convolution layers are applied at the first layer in the TL classifier. The remaining network in the classifier is designed with a FCN considering the computational complexity and accuracy performance. The well-known Bosch traffic dataset is used for training and performance evaluation of the proposed method. The performance of the proposed method is compared with conventional two-staged TL recognition methods in terms of TL candidate detection and recognition performances, hardware requirements, and computational complexity.

The rest of this paper is organized as follows. In Section 2, we describe the proposed TL recognition method. In Section 3, the proposed method is empirically analyzed for various performance metrics, and the performance of the proposed method is compared with the conventional method. Section 4 draws the conclusions.

2. Proposed Traffic Light Recognition Method

In this section, we present a two-staged deep-learning based TL recognition method that consists of candidate detection and classification stages, as shown in Figure 1. TL candidate regions and their positions are extracted in the candidate detection stage. For the classification stage, the candidate regions are discriminated into types of TLs, including background. The following subsections describe the two stages in detail. In addition, the training and inference processes are also described.

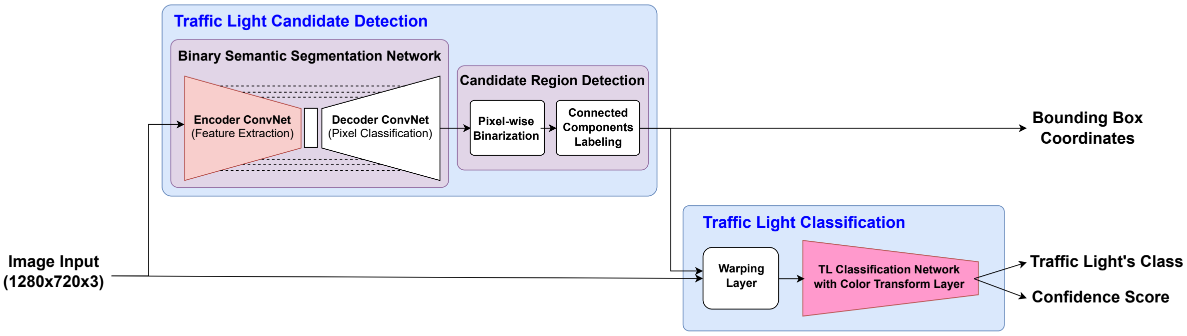

Traffic Light Candidate Detection
Binary Semantic Segmentation Network
Encoder ConvNet (Feature Extraction)
Decoder ConvNet (Pixel Classification)
Candidate Region Detection
Pixel-wise Binarization
Connected Components Labeling
Bounding Box Coordinates
Image Input (1280x720x3)
Traffic Light Classification
Warping Layer
TL Classification Network with Color Transform Layer
Traffic Light's Class
Confidence Score

Figure 1. Proposed two-staged traffic light recognition method.

2.1. *Traffic Light Candidate Detection Stage*

Conventional object detection (OD) methods are not suitable to detect small objects, because they use very deep depth networks with pooling operations for feature extraction [46–50]. For example, Faster-RCNN employs anchor boxes to extract object candidate regions from a feature map. As the feature map is obtained by deep ConvNet, it has too wide a receptive field to reflect the existence of small objects. For the anchor box layer, additional memory is required according to the maximum number of object candidates. In addition, bounding box regression and non-maximum suppression (NMS) [51] are needed to calculate the precise location of a candidate region and to remove overlapping candidate regions, respectively [30]. These operations make it difficult to implement real-time processing.

The main idea of the proposed candidate detection is to employ a pixel-wise semantic segmentation that is applicable to very small objects. The proposed TL candidate detection stage consists of binary semantic segmentation and candidate region detection. Through the binary semantic segmentation, a confidence score is assigned to each pixel of an input image. The confidence score represents the possibility that each pixel belongs to the traffic light region. For the semantic segmentation, an FCN with an encoder-decoder structure can be used. In this work, we apply E-Net [45], which is efficient for both computational complexity and small object segmentation. Hereafter, the E-Net-based binary semantic segmentation is denoted as BSSNet.

In the detection of a candidate region, the bounding box of the region is calculated. The binary image is obtained by thresholding the confidence score of each corresponding pixel from BSSNet. To extract as many TL candidates as possible, all non-zero confidence scores are segmented as candidates. It should be noted that high threshold values may cause valid TLs to be excluded in the classification stage. Then, eight-connected-neighborhood-based connected components labeling (CCL) [52] is applied to the binary image to obtain separate candidate regions. The bounding box coordinates $(x_{min}(i), y_{min}(i), x_{max}(i), y_{max}(i))$ of the i^{th} candidate region are calculated, where $[x_{min}(i)$ and $y_{min}(i)]$ and $[x_{max}(i)$ and $y_{max}(i)]$ are coordinates of the top left and bottom right corners, respectively.

Unlike conventional two-staged ODs, the proposed candidate detection method does not require an anchor box layer and NMS operations. Therefore, the proposed method can be implemented with a relatively small memory and low computational complexity.

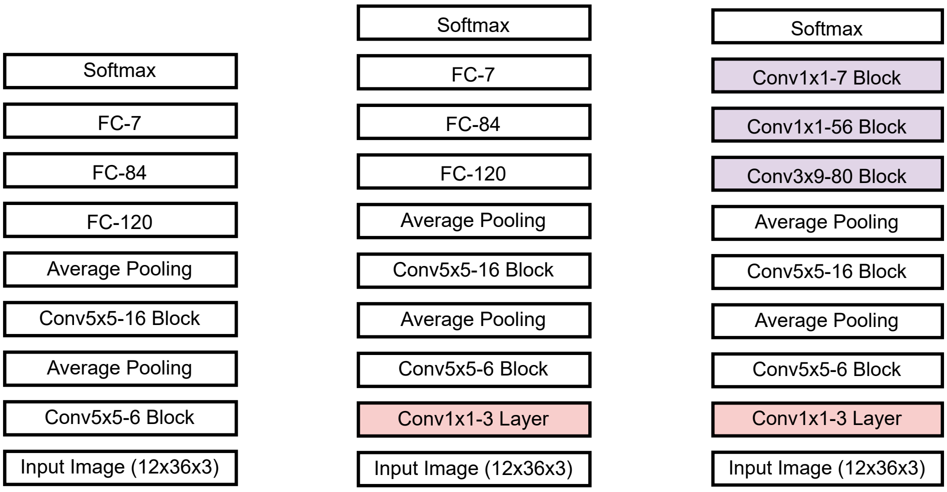
2.2. *Traffic Light Classification Stage*

The TL classification stage classifies the types of traffic lights by using an input image corresponding to a candidate region obtained from the candidate detection stage. The stage consists of a warping layer and a TL classification network. It is observed in our evaluation dataset that most TLs are composed of three lamps. Considering the average width and height of traffic lights as shown in Table 1, the candidate region is cropped from the input image and wrapped to the size of 12 × 36 pixels. 89.01% of TLs belong to small size with less than 32^2. The TL classification network classifies TL candidates into seven types such as red, red-left, green, green-left, yellow, off, and background. Considering the input image size and the number of classes, an LeNet-5-based [53]

TL classification network is designed. Three TL classification networks are proposed and implemented, as shown in Figure 2.

Table 1. Analysis of traffic light size in evaluation dataset.

Traffic Light's Size	Proportion (%)	Width [pixel]			Height [pixel]		
		Min	Max	Mean	Min	Max	Mean
Small (area < 32^2)	89.01	2	40	9.89	4	76	23.81
Non-small ($32^2 \leq$ area)	10.99	15	99	27.39	24	208	62.45
Total	100.00	2	99	**11.14**	4	208	**26.56**



(**a**) TLC1Net (**b**) TLC2Net (**c**) TLC3Net

Figure 2. Proposed traffic light classification networks.

The first network, hereafter referred to as TLC1Net, consists of two convolution blocks and three fully connected (FC) layers, as shown in Figure 2a. Each convolution block is composed of a convolution layer, a batch-normalization layer, and an activation function, in consecutive order. In each convolution block, a convolution layer with K filters of (N × M) is applied, which is denoted as the Conv N × M-K block. Unlike the LeNet-5 [53], the TLC1Net applies zero padding and *ReLU* [54]. In addition, batch normalization is applied between the convolution layer and the activation layer. Average pooling with factor 2 is applied after the first and second convolution blocks. The second proposed classification network, TLC2Net, is designed by adding a convolution layer with three filters of (1 × 1) to TLC1Net, as shown in Figure 2b. The additional layer is applied directly to the three-color channels of input data to perform an effective color space transform. As mentioned previously, a TL appears mainly with four colors such as red, green, yellow, and black. If we apply a color space that distinguishes the colors well, we can improve the classification performance. Although various existing color transforms can be applied before the classification network as in our previous works [40], effective color space transform coefficients are obtained through training processes. No bias weight is applied for the color transform layer. Figure 2c shows the third proposed TL classification network, TLC3Net, that is designed with FCN. As the three FC layers in TLC2Net are replaced by three convolution blocks, TLC3Net is faster than the TLC2Net, while reducing weight parameters. All three proposed TL classification networks have $softmax$ [55] at the end to discriminate the type of TLs based on the confidence scores of the seven classes.

2.3. Training Process

A multi-task training process is carried out for TL candidate detection and classification. In the training task of TL candidate detection, the input data of the binary semantic segmentation network is an RGB color image, and the ground truth (GT) is a corresponding binary image in which only the pixels in the traffic light regions are "1". As $softmax$ is applied as the activation function at the last layer, cross-entropy loss [55] is used as the objective function. The network is trained until a maximum of 2000 epochs. During training, an adaptive moment estimation solver called Adam [56] is applied with a batch size of four, learning rate 10^{-4}, and momentum parameters $\beta_1 = 0.9$, $\beta_1 = 0.999$, and epsilon(ϵ) = 10^{-8}. Early stopping [57,58] is applied with patience parameter 50 of the validation loss minimum.

In the training task of TL classification, a previously-trained result in the binary semantic segmentation network is used. TL candidate regions extracted through the candidate region detection are cropped from the input image and wrapped to the size of 12 × 36 pixels. The resized TL candidate region is used as an input of the TL classification network. At this time, intersection over union (IoU) [59] is calculated by comparing the coordinates between the candidate region and the GT. If IoU is greater than or equal to 0.5 (IoU $\geq$ 0.5) [28–37,41,59], the TL candidate region is trained as the corresponding class of the GT. In contrast, if the IoU is less than 0.5 (IoU < 0.5) or there is no TL, the TL candidate region is trained as background. As the last layer of the TL classification network also uses softmax, cross-entropy loss is applied. The classification network is trained until the maximum of 200 epochs. During training, an Adam is also applied with learning rate 10^{-4} and momentum parameters $\beta_1 = 0.9$, $\beta_1 = 0.999$, and epsilon(ϵ) = 10^{-8}. For TL classification training, the batch size varies depending on the number of TL candidates included in the given input image of driving road scenes. Early stopping is applied with patience parameter 10 of the validation loss minimum.

2.4. Inference Process

The TL candidate region is extracted from the RGB color image through the binary semantic segmentation network and the candidate region detection. The TL candidate region is resized into 12 × 36 size through the warping layer, and is classified into seven classes through the TL classification network. The class with the highest confidence score is finally selected. Except for the background class, the TL recognition outputs the type of TL, bounding box coordinates, and confidence score.

3. Simulation Environments and Performance Results

In this section, we evaluate the performance of the proposed TL recognition as compared with conventional two-staged deep-learning-based OD. For example, Faster R-CNN with inception-resnet -v2 [40] is compared. The performances are evaluated in terms of TL candidate detection and TL recognition. Before analyzing the performance, we briefly describe the dataset and measurement metrics for the performance evaluation in the following subsections.

3.1. Evaluation Dataset

For the simulations, we use an augmented version of the Bosch Small Traffic Lights Dataset [26] that is used in [40]. The evaluation dataset consists of 8144 RGB color images with 1280 × 720 resolution, and contains 17,102 annotated traffic lights. Six types of TLs are included, such as green, red, yellow, green-left, red-left, and off. The training and test datasets consist of 6102 and 2042 images, respectively, selected from 8144 color images. The proportion of training and test datasets of '3:1' are widely adopted throughout literature [60–62]. For performance comparison, we use the same data sets as in [40], which can be referred to [40] for more information. The test dataset is also used for validation.

3.2. Performance Measurement Metrics

TL candidate detection performance is evaluated by three metrics, such as precision, recall, and F-measure. For TL recognition performance, four metrics are used, such as average precision (AP), mean average precision (mAP), overall AP, and overall mAP. In addition, the average processing time is evaluated to verify the speed performance of the proposed TL recognition method. The network size according to the number of weight parameters is also compared.

3.3. TL Candidate Detection Performances

Table 2 shows the detection performances where the proposed BSSNet is applied for two different sizes of input image. In this table, BSSNet-full-size and BSSNet-half-size represent the BSSNet tested on input images of 1280 × 720 and 640 × 360, respectively. For these, BSSNet is trained independently for the two cases. Faster R-CNN with inception-resnet-v2 [40] is also tested for the input image of 1280 × 720 and compared. The performances are listed according to the sizes of the TL, i.e., small (# of pixel $\leq 32^2$) and non-small (# of pixels $> 32^2$) [41]. A false negative indicates the cases in which a TL is not detected, or where the IoU is found to be less than 0.5 (IoU < 0.5). A false positive indicates the erroneous proposals where a background is misclassified as a TL candidate. The bold-marked numbers indicate the top-ranked method.

Table 2. TL candidate detection performances.

Measure Metrics	Faster R-CNN			BSSNet-Full-Size			BSSNet-Half-Size		
	Total	Small	Non-Small	Total	Small	Non-Small	Total	Small	Non-Small
No. of Traffic Lights (GT)	4306	3815	491	4306	3815	491	4306	3815	491
No. of True Positive	3685	3229	456	**4088**	**3597**	**491**	3908	3417	**491**
No. of False Negative	621	586	35	**218**	**218**	**0**	398	398	**0**
No. of False Positive	2619	1933	686	203	139	64	**128**	**99**	**29**
Precision (%)	58.45	62.55	39.93	95.27	96.28	88.47	**96.83**	**97.18**	**94.42**
Recall (%)	85.58	84.64	92.87	**94.94**	**94.29**	**100.00**	90.76	89.57	**100.00**
F-measure (%)	69.46	71.94	55.85	**95.10**	**95.27**	93.88	93.69	93.22	**97.13**

The proposed methods detect TL candidates better than Faster R-CNN in terms of all three metrics, i.e., precision, recall, and F-measure. BSSNet remarkably outperforms the conventional Faster-RCNN from the viewpoint of false positives. In particular, the proposed BSSNet has a relatively small number of false negatives. Note that a false negative has a direct effect on the performance of whole TL recognition system, as the region does not deliver to the TL classification stage. As expected, BSSNet-full-size has slightly better performance than BSSNet-half-size in terms of F-measure in total. This is caused by the fact that BSSNet-full-size detects small size TLs better than BSSNet-half-size, and the number of small size TLs is larger than the number of non-small ones.

3.4. TL Recognition Performances

The final performance of the proposed TL recognition is evaluated in terms of overall mAP and mAP@0.5, as shown in Tables 3 and 4. As mentioned in Section 2.2, three TL classification networks such as TLC1Net, TLC2Net, and TLC3Net are applied and compared. BSSNet-full-size and BSSNet-half-size are applied to the three classification networks, respectively. A candidate region taken from a full-size input image is warped and fed to the classification network for both training and inference processes. In the case of BSSNet-half-size, the bounding box coordinates of the candidate region are scaled up by two, to crop the candidate region from the full-size input image. In the tables, the top-ranked method is marked with bold face.

Table 3. TL recognition performances (overall mAP and overall AP) on test set.

TL Recognition Method	Overall mAP (%)			Overall AP (%)					
	Total	Small	Non-Small	Green	Red	Yellow	Red-Left	Green-Left	Off
Faster R-CNN [40]	20.40	15.85	36.15	33.46	23.81	4.75	34.69	17.59	8.08
BSSNet-full-size & TLC1Net	41.00	34.94	69.09	49.97	35.19	24.66	57.03	51.00	28.12
BSSNet-full-size & TLC2Net	42.96	36.93	72.31	52.07	37.79	26.47	59.04	53.00	29.39
BSSNet-full-size & TLC3Net	**44.50**	**38.98**	**75.52**	**53.91**	**39.16**	**27.82**	**61.20**	**54.36**	**30.52**
BSSNet-half-size & TLC1Net	31.04	25.55	66.53	41.81	25.5	19.34	47.38	38.76	13.46
BSSNet-half-size & TLC2Net	34.22	28.70	70.01	43.18	28.42	21.52	53.21	42.74	16.25
BSSNet-half-size & TLC3Net	36.32	30.69	73.62	44.80	29.60	22.65	54.74	48.02	18.10

Table 4. TL recognition performances (mAP@0.5 and AP@0.5) on test set.

TL Recognition Method	mAP@0.5 (%)			AP@0.5 (%)					
	Total	Small	Non-Small	Green	Red	Yellow	Red-Left	Green-Left	Off
Faster R-CNN [40]	38.48	31.27	57.79	70.56	52.12	8.49	59.11	27.13	13.44
BSSNet-full-size & TLC1Net	65.32	59.04	82.86	85.47	70.64	41.88	74.28	65.89	53.73
BSSNet-full-size & TLC2Net	67.67	62.26	86.65	88.05	72.92	45.71	76.95	67.48	54.93
BSSNet-full-size & TLC3Net	**70.16**	**64.88**	**89.99**	**90.66**	**74.89**	**48.77**	**79.67**	**69.75**	**57.20**
BSSNet-half-size & TLC1Net	50.67	44.94	79.60	76.58	54.43	31.59	63.29	50.36	27.78
BSSNet-half-size & TLC2Net	54.35	48.61	83.41	77.94	57.19	34.38	70.55	55.15	30.87
BSSNet-half-size & TLC3Net	57.73	52.19	87.90	80.55	59.26	36.18	72.45	62.16	35.79

In our previous work [40], three conventional methods, such as Faster R-CNN with inception-Resnet-v2 [63], Faster R-CNN with Resnet-101 [63], and R-FCN with Resnet-101 [63], are compared. Since the first method shows the best performance, it is used for the performance comparison in this paper and denoted as 'Faster R-CNN'. As compared to the Faster R-CNN method, the proposed TL recognition methods show significantly improved performances. In particular, the proposed TL recognition with BSSNet-full-size and TLC3Net improves performances by 24.1% in Overall mAP and by 31.68% in mAP@0.5, as compared with the conventional Faster R-CNN. It is observed that false positives are well-classified into background in the proposed TL classification networks. Among the proposed three classification networks, TLC3Net shows the best performance. TLC2Net produces improvements of 1.96% in overall mAP and 2.35% in mAP@0.5 over TLC1Net. It implies that the added convolution layer with (1 × 1) filters is particularly useful to extract the main color components of TL. TLC3Net produces improvements of 1.54% in overall mAP and 2.49% in mAP@0.5 over TLC2Net. It shows that the FCN is more useful than the FC layer for both complexity and performance. BSSNet-half-size-based TL recognition methods have also higher performance than Faster R-CNN at a much smaller computational complexity.

3.5. Performance Shift Analysis of TL Recognition

In this Section, the performance variations are examined in terms of the ratio of training to test datasets, data swapping between the two datasets for a given ratio, and different database. For the first two evaluations, the proposed BSSNet-full-size and TLC3Net, which shows the best performance in Section 3.4, is used for the Bosch TL dataset. To know the performance variation by different database, the proposed BSSNet-full-size and TLC3Net is evaluated for the LISA Traffic Light Database [3,64]. The performance measures, mAP@0.5 is used for the analysis on performance variations.

For the evaluation of performance variations by ratios of training to test datasets, the selected ratios are 1:1, 3:1, and 5:1. Note that the total dataset is composed of 8144 images with 17,102 traffic lights from the Bosch TL dataset. In case of the ratio of 1:1, training and test datasets are composed of 4072 and 4072 images, respectively. For the ratios of 3:1 and 5:1, '6102 and 2042' and '6783 and

1361′ images. For each simulation, 50% images in test dataset are randomly selected and swapped with training images. For each ratio of training to test dataset, the simulations are conducted by five times with the different swapped images. Average recognition performances are provided in Figure 3. The results implicate that the ratio of 3 to 1 gives the best performance, compared with other ratios but the performance variations are marginal. Performance shift by ratio dataset is only 1.99% in mAP@0.5, while the performance improvement of the proposed method over conventional method in mAP@0.5 is 31.55%. This implies that the performance shift is negligible with respect to the selection of swapped images. These tolerances to the variation of the ratios and image selections for training and test datasets come from the fact that the amount of data used in training is enough to train the small number of classes, i.e., the more than 4000 images for only six classes.

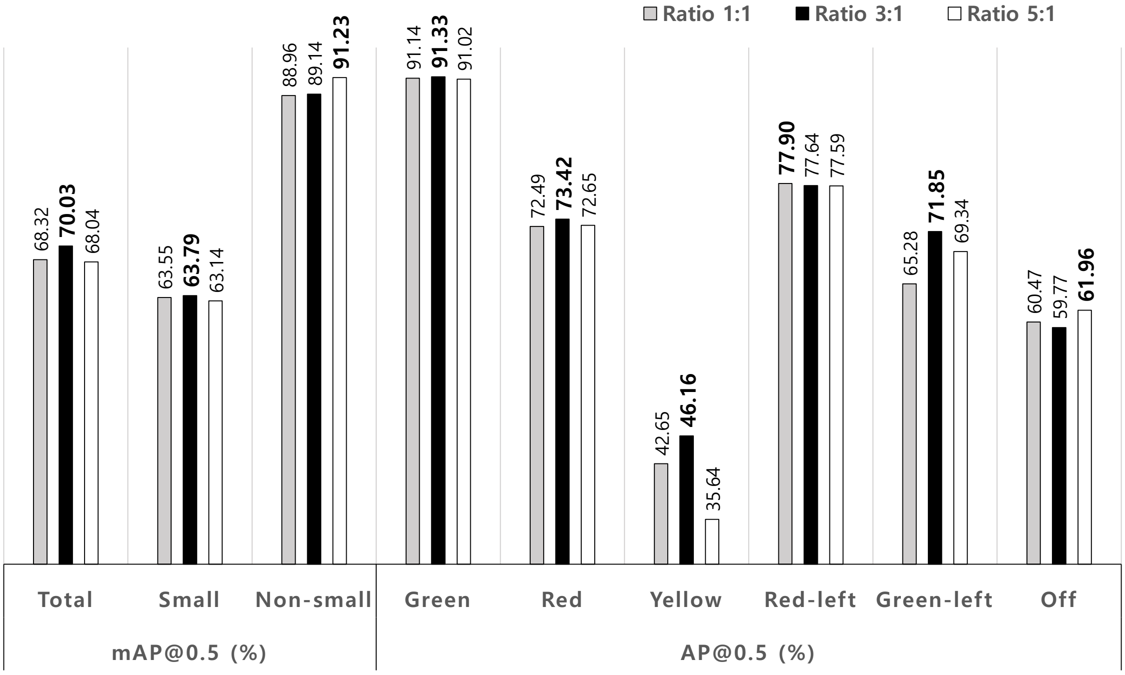


Figure 3. Average TL recognition performances according to ratio of training to test datasets.

To evaluate the bias effect by selecting training data from the given total dataset, different proportions of test dataset are swapped with training dataset for a given ratio of training to test dataset, 3:1. For instance, the ’50% (1021)’ in Figure 4 means that 1021 images are swapped between training and test datasets.The swapping images are randomly selected for both datasets. For each proportion, the simulations are conducted by five times with the different swapped images. Figure 4 shows that the proposed method gives very robust performance to the selection of training and test datasets.

To evaluate validity to different database, the proposed and conventional methods are trained and tested on LISA Traffic Light Database [3,64]. The LISA database is obtained in the various environments during the daytime. It consists of 20,089 RGB color images with 55,536 annotated traffic lights. Six classes of TLs are go, go-left, warning, warning-left, stop, and stop-left. The database provides separate training and test datasets. The training and test datasets consist of 12,775 and 7314 images, respectively. Table 5 shows that the proposed method gives improvement in mAP@0.5 by 33.97%, compared with the conventional method. From the results on Bosch database in Section 3.4, the improvement in mAP@0.5 is 31.68%. The proposed method produces the similar amount of improvements for both databases. This result shows that the proposed method has the validity to different database.

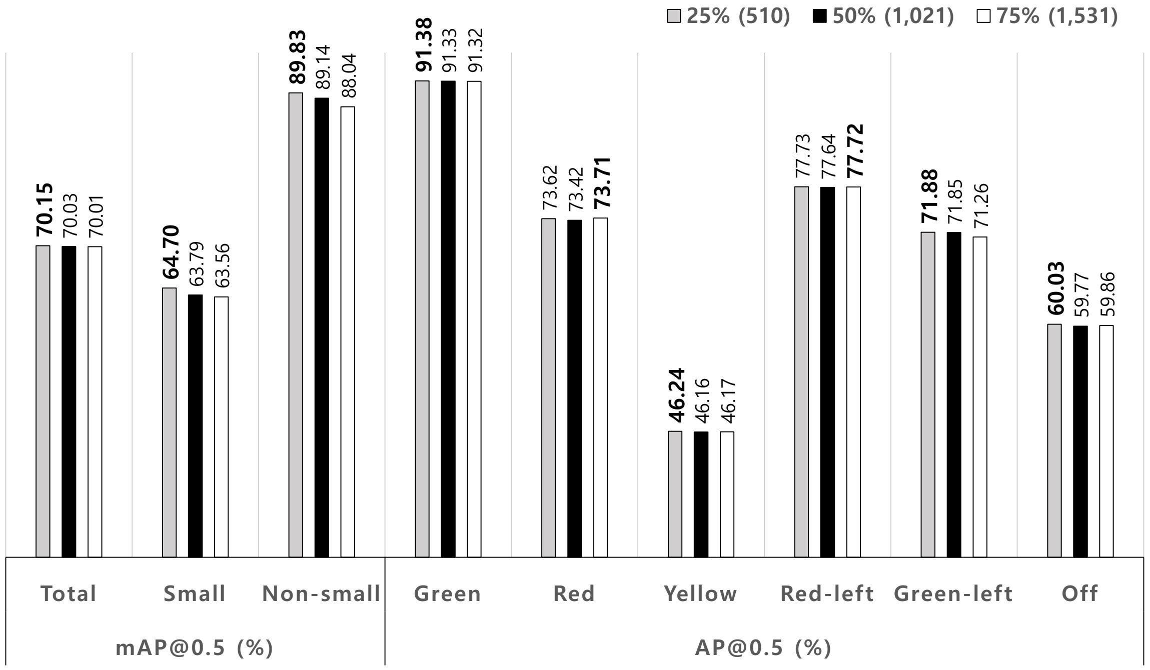


Figure 4. Average TL recognition performances according to dataset shift.

Table 5. TL recognition performances on test dataset in LISA database.

TL Recognition Method	**mAP@0.5 (%)**			**AP@0.5 (%)**					
	Total	**Small**	**Non-Small**	**Go**	**Go-Left**	**Warning**	**Warning-Left**	**Stop**	**Stop-Left**
Faster R-CNN	44.34	40.05	48.63	53.22	34.84	42.97	33.97	52.26	43.78
BSSNet-full-size & TLC3Net	**78.31**	**65.16**	**91.47**	**83.24**	**70.19**	**83.41**	**69.86**	**88.12**	**75.07**

3.6. Hardware Requirements

To analyze hardware requirements, we compare the network size according to the total number of weight parameters. We also analyze the size for each sub-network of the proposed TL recognition method. Table 6 shows the network size in megabytes (MB). The proposed TL recognition method requires approximately less than 1% of the network size of Faster-RCNN. In addition, the proposed recognition methods require slightly different network sizes depending on the TL classification network. The proposed BSSNet has the same number of weight parameters, even if the input size of the BSSNet is changed. Thus, BSSNet-x-size denotes both BSSNet-full-size and BSSNet-half-size in Table 6. The detailed network size for each sub-network in the proposed TL recognition method is summarized in Table 7. TLC2Net has very slightly larger network size than TLC1Net, as the color transform layer is added to the TLC1Net. TLC3Net has a relatively small size, because the FC layers are replaced by the FCN.

Table 6. Hardware requirements according to the TL recognition.

TL Recognition Method	**Network Size [MB]**	**Comparison**
Faster R-CNN	242.2	1x
BSSNet-x-size & TLC1Net	2.04	0.0084x
BSSNet-x-size & TLC2Net	2.05	0.0085x
BSSNet-x-size & TLC3Net	**1.97**	**0.0081x**

Table 7. Detail network size for each sub-network of functional stage.

Functional Stage	Sub-Network	# of Weight Parameters	Network Size [MB]
TL Candidate Detection	BSSNet-full-size	366,482	1.76
	BSSNet-half-size	366,482	1.76
TL Classification	TLC1Net	65,807	0.28
	TLC2Net	65,816	0.29
	TLC3Net	42,687	0.21

3.7. Computational Complexity

Computational complexity is evaluated by average processing time, as shown in Table 8. The inference processing time is measured on one Intel Core i7-6850K 3.60 GHz CPU and one NVIDIA Titan X Pascal GPU. Through simulations, it is observed that the proposed recognition methods have the same average processing time regardless of the TL classification network. This is caused by the fact that all three classification networks have almost the same number of weight parameters, as mentioned in Section 3.6. Then, the notation TLCxNet is used in Table 8.

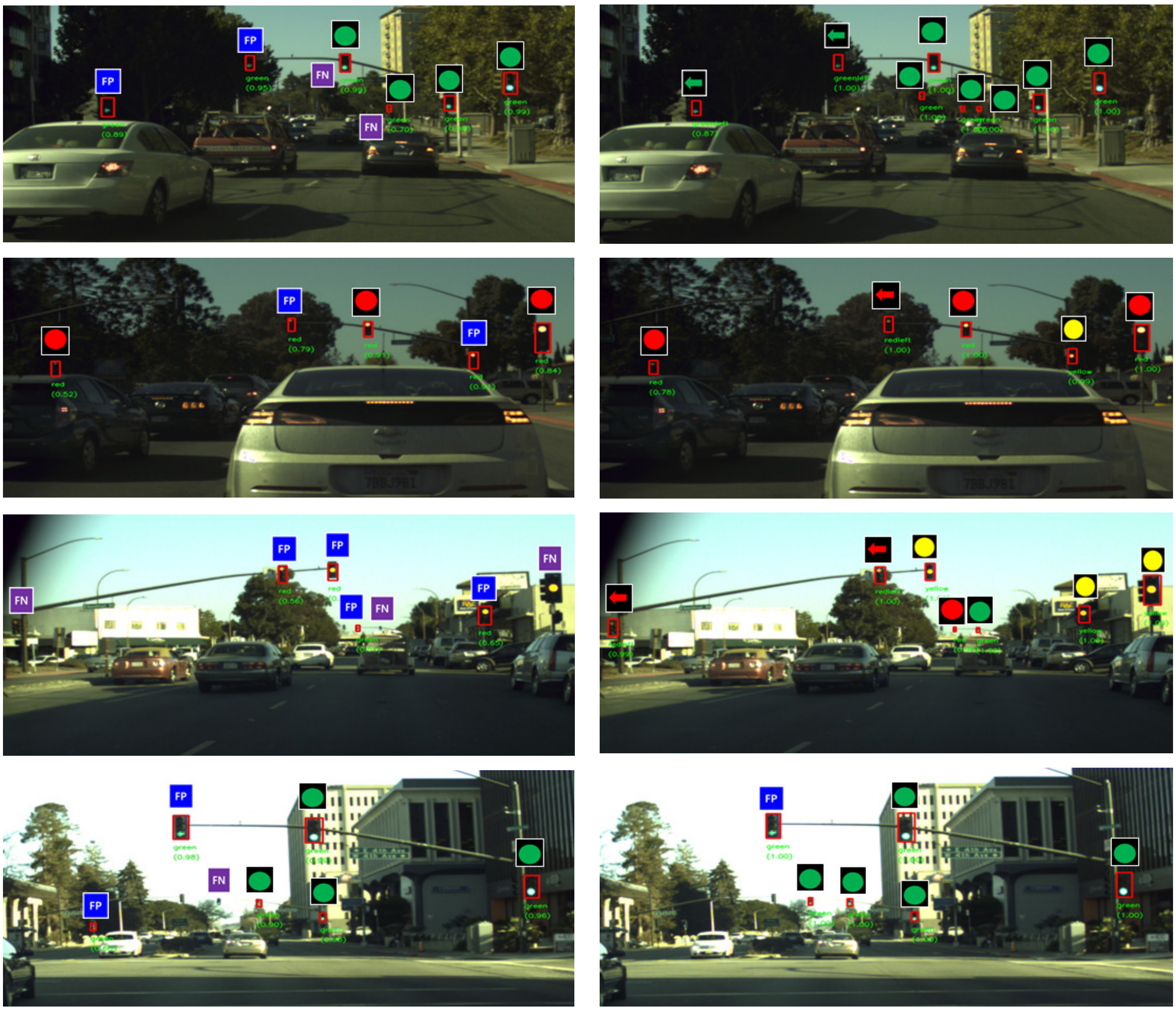
Table 8. Computational complexity according to TL recognition.

TL Recognition Method	Average Processing Time		Comparison
	[ms]	[fps]	
Faster R-CNN	525	1.90	1x
BSSNet-full-size & TLCxNet	96	10.42	5.47x
BSSNet-half-size & TLCxNet	**34**	**29.41**	**15.44x**

The proposed TL recognition method with BSSNet-full-size is 5.47 times faster than Faster R-CNN. In the proposed methods, it takes the same average processing time for the candidate region detection (7 ms on CPU), warping operation (1 ms on CPU), and TL classification (1 ms on GPU), regardless of the input image size of BSSNet. It only takes different processing times for BSSNet-full-size (87 ms on GPU) and BSSNet-half-size (25 ms) because of the different sizes of the input image. As BSSNet is dominant factor for processing time, the proposed TL recognition method with BSSNet-half size can be implemented in real-time with the sacrifice of a minor decrease in recognition performance.

3.8. TL Recognition Examples

Figure 5 shows the TL recognition examples of the proposed TL recognition method with the BSSNet-full-size and TLC3Net. Faster R-CNN examples are also shown. A true positive is indicated by the corresponding six types of TL symbol. False positive and false negative cases are indicated by a blue rectangle with FP and a purple rectangle with FN, respectively. As shown in Figure 5a, the Faster R-CNN has eleven true positives, nine false positives, and six false negatives in four images. Figure 5b shows that the proposed TL recognition method has twenty-five true positives and one false positive. As shown in the Figure 5, the proposed method has much better TL recognition performance than the conventional method. The first row in Figure 5 shows that the Faster R-CNN does not detect two small TLs (denoted by FN) and mis-classifies two TLs (denoted by FP). On the contrary, the proposed method effectively recognizes small TLs. These trends can be observed in other rows in Figure 5. The result shown in the last row reveals that the proposed and conventional methods mis-classify ′green-left′ into ′green′ (denoted by FP). The percentages of ′green-left′ and ′green′ in dataset are 1.67% and 48.43%, respectively. This implicates that the ′green-left′ data need to be supplemented in the dataset. One interesting result is that the proposed method gives a higher TL confidence score than the conventional method, even when Faster R-CNN also recognizes a TL.



(**a**) Faster R-CNN.

(**b**) Proposed method with TCL3Net.

Figure 5. TL recognition examples of the Faster R-CNN and the proposed method.

4. Conclusions

In this study, we propose a two-staged deep-learning-based traffic light recognition method that consists of candidate detection and classification stages. To efficiently reduce the number of weight parameters and computational complexity, a semantic segmentation technique and a fully convolutional network (FCN) are applied. A binary-semantic segmentation network is proposed to detect small-size traffic lights. We also propose a novel traffic light classification network including a convolution layer with three filters of (1 × 1). The simulation results show that the proposed traffic light recognition method outperforms the conventional Faster R-CNN in terms of recognition performance, and it remarkably reduces the computational complexity and hardware requirements. The traffic light recognition method achieves up to 44.5% in overall mAP and 70.16% in mAP@0.5. Especially, the empirical results show that the proposed method gives great improvement for the detection and recognition of small TLs. The proposed method can also be implemented in real-time processing with the sacrifice of a minor decrease in recognition performance. This framework can be a powerful network design guideline for the detection and recognition of small objects like traffic lights. Further research is to improve the recognition performance for "green-left" and "yellow" TLs, which occur in very short period of time.

Author Contributions: H.-K.K. performed conception, design, methodology, data collection, implementation, testing, validation, and writing; K.-Y.Y. helped in conception, design, methodology, data analysis, and writing; J.H.P. guided the research work; H.-Y.J. supervised the whole research project.

Funding: This work was supported in part by Basic Science Research Programs through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant NRF-2018R1A2B6005105), in part by the Brain Korea 21 Plus Program (No. 22A20130012814) funded by National Research Foundation of Korea (NRF), and in part by Ministry of Science, ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2019-2016-0-00313) supervised by the Institute for Information & communications Technology Promotion (IITP).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. On-Road Automated Driving Committee. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J.* **2014**, *3016*, 1–16. 10.4271/J3016_201401. [CrossRef]
2. Diaz, M.; Cerri, P.; Pirlo, G.; Ferrer, M.A.; Impedovo, D. A Survey on Traffic Light Detection. In Proceedings of the New Trends in Image Analysis and Processing, Genoa, Italy, 7–8 September 2015; pp. 201–208.
3. Jensen, M.B.; Philipsen, M.P.; Møgelmose, A.; Moeslund, T.B.; Trivedi, M.M. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1800–1815. [CrossRef]
4. Fairfield, N.; Urmson, C. Traffic light mapping and detection. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5421–5426.
5. Levinson, J.; Askeland, J.; Dolson, J.; Thrun, S. Traffic light mapping, localization, and state detection for autonomous vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5784–5791.
6. John, V.; Yoneda, K.; Qi, B.; Liu, Z.; Mita, S. Traffic light recognition in varying illumination using deep learning and saliency map. In Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems, Qingdao, China, 8–11 October 2014; pp. 2286–2291.
7. Philipsen, M.P.; Jensen, M.B.; Trivedi, M.M.; Møgelmose, A.; Moeslund, T.B. Ongoing work on traffic lights: Detection and evaluation. In Proceedings of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance, Karlsruhe, Germany, 25–28 August 2015; pp. 1–6.
8. Barnes, D.; Maddern, W.; Posner, I. Exploiting 3D semantic scene priors for online traffic light interpretation. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Seoul, South Korea, 28 June–1 July 2015; pp. 573–578.
9. Hosseinyalmdary, S.; Yilmaz, A. Traffic Light Ddetection Using Conic Section Geometry. *ISPRS J. Photogramm. Remote. Sens.* **2016**, *3*, 191–200. [CrossRef]
10. Jang, C.; Cho, S.; Jeong, S.; Suhr, J.K.; Jung, H.G.; Sunwoo, M. Traffic light recognition exploiting map and localization at every stage. *Expert Syst. Appl.* **2017**, *88*, 290–304. [CrossRef]
11. Fregin, A.; Müller, J.; Dietmayer, K. Three ways of using stereo vision for traffic light recognition. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 430–436.
12. Wang, J.G.; Zhou, L.B. Traffic light recognition with high dynamic range imaging and deep learning. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1341–1352. [CrossRef]
13. De Charette, R.; Nashashibi, F. Traffic light recognition using image processing compared to learning processes. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 333–338. [CrossRef]
14. Yu, C.; Huang, C.; Lang, Y. Traffic light detection during day and night conditions by a camera. In Proceedings of the IEEE 10th International Conference on Signal Processing, Beijing, China, 24–28 October 2010; pp. 821–824. [CrossRef]
15. De Charette, R.; Nashashibi, F. Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates. In Proceedings of the IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 358–363. [CrossRef]
16. Kim, H.K.; Park, J.H.; Jung, H.Y. Effective traffic lights recognition method for real time driving assistance system in the daytime. *Int. J. Electr. Comput. Eng.* **2011**, *5*, 1429–1432.

17. Siogkas, G.; Skodras, E.; Dermatas, E. Traffic Lights Detection in Adverse Conditions using Color, Symmetry and Spatiotemporal Information. In Proceedings of the International Conference on Computer Vision Theory and Applications, Rome, Italy, 24–26 February 2012; pp. 620–627.
18. Kim, H.K.; Shin, Y.N.; Kuk, S.G.; Park, J.H.; Jung, H.Y. Night-time traffic light detection based on svm with geometric moment features. *Int. J. Comput. Inf. Eng.* **2013**, *7*, 472–475.
19. Jang, C.; Kim, C.; Kim, D.; Lee, M.; Sunwoo, M. Multiple exposure images based traffic light recognition. In Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 8–11 June 2014; pp. 1313–1318.
20. Kim, H.K.; Park, J.H.; Jung, H.Y. Vision based Traffic Light Detection and Recognition Methods for Daytime LED Traffic Light. *IEMEK J. Embed. Syst. Appl.* **2014**, *9*, 145–150. [CrossRef]
21. Diaz-Cabrera, M.; Cerri, P.; Medici, P. Robust real-time traffic light detection and distance estimation using a single camera. *Expert Syst. Appl.* **2015**, *42*, 3911–3923. [CrossRef]
22. Almagambetov, A.; Velipasalar, S.; Baitassova, A. Mobile standards-based traffic light detection in assistive devices for individuals with color-vision deficiency. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1305–1320. [CrossRef]
23. Shi, Z.; Zou, Z.; Zhang, C. Real-time traffic light detection with adaptive background suppression filter. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 690–700. [CrossRef]
24. Saini, S.; Nikhil, S.; Konda, K.R.; Bharadwaj, H.S.; Ganeshan, N. An efficient vision-based traffic light detection and state recognition for autonomous vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 606–611.
25. Lee, G.G.; Park, B.K. Traffic light recognition using deep neural networks. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 8–10 January 2017; pp. 277–278.
26. Behrendt, K.; Novak, L.; Botros, R. A deep learning approach to traffic lights: Detection, tracking, and classification. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 1370–1377.
27. Li, X.; Ma, H.; Wang, X.; Zhang, X. Traffic light recognition for complex scene with fusion detections. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 199–208. [CrossRef]
28. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
29. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
31. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1605.06409.
32. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
33. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242.
34. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
35. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
37. Fu, C.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
38. Jensen, M.B.; Nasrollahi, K.; Moeslund, T.B. Evaluating state-of-the-art object detector on challenging traffic light data. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 882–888.
39. Müller, J.; Dietmayer, K. Detecting Traffic Lights by Single Shot Detection. *arXiv* **2018**, arXiv:1805.02523.

40. Kim, H.K.; Park, J.H.; Jung, H.Y. An Efficient Color Space for Deep-Learning Based Traffic Light Recognition. *J. Adv. Transp.* **2018**, *2018*, 2365414. [CrossRef]
41. 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 Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
42. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
43. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv* **2015**, arXiv:1511.00561.
44. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
45. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv* **2016**, arXiv1606.02147.
46. Menikdiwela, M.; Nguyen, C.; Li, H.; Shaw, M. CNN-based small object detection and visualization with feature activation mapping. In Proceedings of the 2017 International Conference on Image and Vision Computing New Zealand, Christchurch, New Zealand, 4–6 December 2017; pp. 1–5.
47. Meng, Z.; Fan, X.; Chen, X.; Chen, M.; Tong, Y. Detecting Small Signs from Large Images. *arXiv* **2017**, arXiv1706.08574.
48. Ren, Y.; Zhu, C.; Xiao, S. Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN. *Appl. Sci.* **2018**, *8*, 813. [CrossRef]
49. Hu, G.X.; Yang, Z.; Hu, L.; Huang, L.; Han, J.M. Small Object Detection with Multiscale Features. *Int. J. Digit. Multimed. Broadcast.* **2018**, *2018*, 4546896. [CrossRef]
50. Truong, T.D.; Nguyen, V.T.; Tran, M.T. Lightweight Deep Convolutional Network for Tiny Object Recognition. In Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods, Funchal, Madeira, Portugal, 16–18 January 2018; pp. 675–682.
51. Rothe, R.; Guillaumin, M.; Van Gool, L. Non-maximum suppression for object detection by passing messages between windows. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 290–306.
52. Dillencourt, M.B.; Samet, H.; Tamminen, M. A general approach to connected-component labeling for arbitrary image representations. *J. ACM* **1992**, *39*, 253–280. [CrossRef]
53. LeCun, Y.; Jackel, L.; Bottou, L.; Brunot, A.; Cortes, C.; Denker, J.; Drucker, H.; Guyon, I.; Muller, U.; Sackinger, E.; et al. Comparison of learning algorithms for handwritten digit recognition. In Proceedings of the International Conference on Artificial Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 60, pp. 53–60.
54. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
55. Kiviluoto, K.; Oja, E. Softmax-network and S-Map-models for density-generating topographic mappings. In Proceedings of the 1998 IEEE International Joint Conference on Neural Networks, Anchorage, AK, USA, 4–9 May 1998, Volume 3 ; pp. 2268–2272. . [CrossRef]
56. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv1412.6980.
57. Prechelt, L. Automatic early stopping using cross validation: Quantifying the criteria. *Neural Netw.* **1998**, *11*, 761–767. [CrossRef]
58. Caruana, R.; Lawrence, S.; Giles, C.L. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2001; pp. 402–408.
59. Rahman, M.A.; Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 12–14 December 2016; pp. 234–244.
60. Kostek, B.; Wójcik, J.; Szczuko, P. Automatic Rhythm Retrieval from Musical Files. In *Transactions on Rough Sets IX*; Springer: Berlin, Heidelberg/Germany, 2008; pp. 56–75. [CrossRef]

61. Tian, Y.; Li, X.; Wang, K.; Wang, F. Training and testing object detectors with virtual images. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 539–546. [CrossRef]
62. Jin, X.; Sun, X.; Zhang, X.; Sun, H.; Xu, R.; Li, X.; Sun, N. Synthesizing Virtual-Real Artworks Using Sun Orientation Estimation. In Proceedings of the International Symposium on Artificial Intelligence and Robotics, Nanjing, China, 24–25 November 2018; pp. 51–58.
63. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3296–3305.
64. Philipsen, M.P.; Jensen, M.B.; Mogelmose, A.; Moseslund, T.; Trivedi, M.M. Learning based traffic light detection: Evaluation on challenging dataset. In Proceedings of the 18th IEEE Intelligent Transportation Systems Conference, Las Palmas, Spain, 15–18 September 2015.