Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 2365414, 12 pages
https://doi.org/10.1155/2018/2365414

Research Article

An Efficient Color Space for Deep-Learning Based Traffic Light Recognition

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

[1] *Multimedia Signal Processing Group, Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, Republic of Korea*
[2] *Nonlinear Dynamics Group, Department of Electrical Engineering, Yeungnam University, Gyeongsan 38544, Republic of Korea*

Correspondence should be addressed to Ho-Youl Jung; hoyoul@yu.ac.kr

Received 21 May 2018; Accepted 19 November 2018; Published 6 December 2018

Guest Editor: Mihai Dimian

Traffic light recognition is an essential task for an advanced driving assistance system (ADAS) as well as for autonomous vehicles. Recently, deep-learning has become increasingly popular in vision-based object recognition owing to its high performance of classification. In this study, we investigate how to design a deep-learning based high-performance traffic light detection system. Two main components of the recognition system are investigated: the color space of the input video and the network model of deep learning. We apply six color spaces (RGB, normalized RGB, Ruta's RYG, YCbCr, HSV, and CIE Lab) and three types of network models (based on the Faster R-CNN and R-FCN models). All combinations of color spaces and network models are implemented and tested on a traffic light dataset with 1280×720 resolution. Our simulations show that the best performance is achieved with the combination of RGB color space and Faster R-CNN model. These results can provide a comprehensive guideline for designing a traffic light detection system.

1. Introduction

Over the past few years, various advanced driving assistance system (ADAS) have been developed and commercialized. In particular, most automotive companies are now doing their best to launch autonomous vehicles as soon as possible. According to the society of automotive engineers (SAE), the international standard defining the six levels of driving automation requires the autonomous driving to achieve level 3 and higher [1]. Obviously, the traffic light recognition is an essential task for ADAS as well as for autonomous vehicle.

For traffic light recognition, various methods have been proposed. These can be analyzed from three aspects such as color space, feature extraction, and verification/classification. Different color spaces, namely, gray scale [2, 3], RGB [4, 5], normalized RGB [6], Ruta's RGB [7], YCbCr [8, 9], HSI [10], HSV [11, 12], HSL [13], and CIE Lab [14], have been used. Moreover, some studies [15–18] have used more than one color spaces. For feature extraction, Haralick's circularity measure [19], Sobel edge detection [20], circle Hough transform [21], 2D Gabor wavelets [22], Haar-likes [23], histogram of oriented gradients (HOG) [24], and geometric features [25] have been applied. For verification/classification, various conventional classifiers have been used, e.g., k-means clustering [26], template matching [27], 2D independent component analysis (ICA) [28], linear discriminant analysis (LDA) [29], decision-tree classifier, k-nearest neighbor (kNN) classifier [30], adaptive boosting algorithm (Adaboost) [31], and support vector machine (SVM) [32]. Recently, some basic deep-learning networks such as LeNet [33], AlexNet [34], and YOLO [35, 36] have been applied to traffic light recognition. Other approaches using visual light road-to-vehicle communication have been developed. LED-typed traffic lights broadcast the information, then photo-diode [37, 38] or high-frame-rate image sensor [39, 40] receives the optical signal. In this paper, we mainly focus on vision-based traffic light recognition using deep-learning.

In the last couple of years, deep-learning has achieved a remarkable success in various artificial intelligence research areas. In particular, deep-learning has become very popular in vision-based object recognition due to its high performances of classification. One of the first advances is OverFeat that applies the convolutional neural network (CNN) [34] to multiscale sliding window algorithm [41]. Girshick et al. proposed a region with CNN (R-CNN), which achieves up to almost 50% improvement on the object detection performance [42]. In R-CNN, object candidate regions are detected and features are extracted using CNN, while objects are classified using SVM. Girshick proposed a Fast R-CNN, which uses selective search to generate object candidates and applies fully connected neural network to classify the objects [43]. However, the selective search algorithm slows down the object detection system performance. Redmon et al. proposed YOLO, which uses a simple CNN approach to achieve real-time processing by enhancing detection accuracy and reducing computational complexity attaining [35]. Ren et al. proposed a Faster R-CNN which replaces the selective search by region proposal network (RPN) [44]. The RPN is a fully convolutional network that simultaneously predicts the object bounds and object/objectless scores at each position. This method makes it possible to implement the end-to-end training. Recently, two notable deep-learning network models were proposed, single shot detector (SSD) and region-based fully convolutional networks (R-FCN). SSD uses multiple sized convolutional feature maps to achieve a better accuracy and higher speed than YOLO [45]. R-FCN is a modified version of Faster R-CNN, which consists of only convolutional networks [46]. It is to be noted that the feature extraction is included in deep-learning detection network in the cases of Fast R-CNN, YOLO, Faster R-CNN, SSD, and R-FCN frameworks. The above-mentioned deep-learning methods have been widely applied to detect objects such as vehicle and pedestrian [47–51]. However, only a few deep-learning based network models have been applied to traffic light detection system [52–55].

From the viewpoints of color representation, various color spaces of input video data have been used in conventional traffic light recognition methods. However, only a few color spaces have been applied in deep-learning based methods. Because color information plays an important role in the performance of traffic light detection, it is necessary to select the color space carefully in deep-learning based methods. In this study, we focus on how to design a high-performance deep-learning based traffic light recognition system. To find color space most suitable to deep-learning based traffic light recognition, six color spaces such as RGB, normalized RGB, Ruta's RYG, YCbCr, HSV, and CIE Lab are investigated. For deep-learning network models, three models based on the Faster R-CNN and R-FCN are applied. All combinations of color spaces and network models are implemented and compared.

The rest of this paper is organized as follows. Second section discusses the previous research works on traffic light detection system. In third section, we describe various color spaces and deep-learning network models. All combinations of color spaces and network models have been designed in this study. In fourth section, we explain the configurations such as parameter and data set for the performance evaluation. Fifth section presents the simulation results. Final section draws the conclusions.

2. Related Works

In this section, we briefly introduce the work done so far on traffic light detection. These works are categorized into two groups, namely, deep-learning based and conventional classification methods, depending on whether the deep-learning is used or not. They are investigated mainly from the viewpoints of color representation and verification/classification. The analysis is summarized in Table 1.

2.1. Conventional Classification Based Methods. In general, conventional traffic light recognition methods mainly consist of two steps, candidate detection, and classification. Various color representations have been used. Charette and Nashashibi did not use any color information [2, 3]. They proposed to use the gray-scale image as input data. After the top-hat morphological filtering, adaptive template matching with geometry and structure information was applied to detect the traffic lights. Park and Jeong used color extraction and k-means clustering for candidate detection [4]. The average and standard deviation of each component in RGB color space were then calculated and used. Here, Haralick's circularity was used for verification. Yu et al. used the difference of each pair of components in RGB space to extract the dominant color [5]. They applied region growing and segmentation for candidate detection. For verification, the information of shape and position was used. Omachi and Omachi used normalized RGB for color segmentation [6]. The edge detection and circle Hough transform were applied for verification of traffic light. Kim et al. used Ruta's RGB based color segmentation for the detection of traffic light candidates at night [7]. Some geometric and stochastic features were extracted and used in SVM classifier. Kim et al. used YCbCr color-based thresholding and shape filtering for candidate detection [8]. Here, Haar-like features and Adaboost were used for classification. Kim et al. used YCbCr color segmentation for candidate detection [9]. Here, candidate blobs with red and green lights were detected by thresholding Cb and Cr components. Various shape and modified Haar-like features were extracted and used in decision-tree classifier. Siogkas et al. used the CIE Lab color space [14], where the multiplications of L and a components (RG), and L and b components (YB) are used to enhance the discrimination of red and green regions. They used fast radial symmetry transform and persistency to identify the color of traffic lights. Cylindrical color spaces such as HSI, HSV, and HSL have also been used [10–13]. Hwang et al. used HSI color-based thresholding, morphological filtering, and blob labeling for candidate detection [10]. For verification, they used convolution of the candidate region with Gaussian mask using existence-weight map. HSV color space was also used, where the histograms of hue and saturation components are used for candidate extraction [11]. Probabilistic template matching was applied for classification. Recently, it has been reported that the detection performance of traffic lights can be improved by using the 3D geometry map that are prebuild from GPS, INS/IMU, and range sensors such as stereo camera or 2D/3D range lidar [12, 13]. Jang et al. used Haar-like feature based Adaboost with 3D map information for candidate

Table 1: Color space and verification/classification used in previous traffic light detection.

Ref. #	Color space	Verification / Classification
[2], [3]	Gray-scale	Template matching
[4]	RGB	K-means clustering, Circularity check
[5]	RGB	Region growing, Color segmentation
[6]	Normalized RGB	Color segmentation, Circle Hough transform
[7]	Ruta's RGB	SVM
[8]	YCbCr	Adaboost
[9]	YCbCr	Decision-tree classifier
[10]	HSI	Gaussian mask, Existence-Weight Map
[11]	HSV	Template matching
[14]	CIE Lab	Fast radial symmetry transform
[12]	HSV	SVM
[13]	HSL	SVM
[15]	Normalized RGB, RGB	Color clustering
[16]	Normalized RGB, RGB	Fuzzy logic clustering
[17]	RGB, YCbCr	Nearest neighbor classifier
[18]	RGB, HSV	LDA, kNN, SVM
[53]	CIE Lab	SVM, LeNet, AlexNet
[52]	HSV	SVM, Simple CNN
[54]	RGB	YOLO v1
[55]	RGB	YOLO 9000

detection [12]. HOG and HSV color histogram were applied to SVM classifier. Moreover, traffic light candidates are detected by using HOG features based linear SVM classifier with the uncertainty of 3D prior that constrains the search regions [13]. For classification, image color distribution in HSL color space is used.

Some researchers used two color spaces [15–18]. Omachi and Omachi used RGB and normalized RGB color spaces to find candidates, and circle Hough transform was applied for verification [15]. Combination of RGB and normalized RGB was also used for color segmentation based on fuzzy logic clustering, where some geometric and stochastic features were used as primary clues to discriminate traffic lights from others [16]. Cai et al. used RGB and YCbCr color spaces for candidate extraction and classification, respectively [17]. Gabor wavelet transform and ICA based features were extracted and applied to the nearest neighbor classifier. Furthermore, red, yellow and green traffic light regions are detected by thresholding based HSV color segmentation and geometrical features [18]. HOG features were extracted in RGB space and used to determine whether arrow sign is on the light or not. Three different classification algorithms such as LDA, kNN, and SVM were applied, respectively.

Since the selection of color space plays the most important role in traffic light detection performance, past studies have explored all the possible options. Clearly, it is important to identify the best among all these color spaces.

2.2. Deep-Learning Based Methods. Deep-learning has been also used for traffic light detection and classification [52–55]. At first, deep learning was applied only in the classification of traffic lights, where candidates were detected by conventional method [52, 53]. Saini et al. used HSV color space-based color segmentation, aspect ratio, and area-based analysis and maximally stable extremal region (MSER) to localize the candidates [52]. HOG features and SVM were used for verification, whereas simple CNN was used for classification. Lee and Park used CIE Lab color space-based segmentation to find the candidate regions [53]. To reduce false regions, they use SVM with size, aspect ratio, filling ratio and position. The classification was performed by two cascaded CNN which consists successively of LeNet and AlexNet. LeNet quickly differentiates between traffic lights and background. AlexNet classifies traffic light types. Recently, deep-learning has been applied both to candidate detection and classification. Behrendt et al. [54] and Jensen et al. [55] applied YOLO-v1 [35] and YOLO-9000 [36] for traffic light detection/classification.

As discussed, only a few color spaces have been applied in deep-learning based traffic light detection. Because color information plays an important role in the performance of detection, it is necessary to select the color space carefully. It is also required to apply more sophisticated and efficient deep-learning network models to traffic light detection and classification.

3. Deep-Learning Based Traffic Light Detection

In this section, we present a deep-learning based traffic light detection system that consists of the preprocessing, deep-learning based detection, and postprocessing as shown in Figure 1. In preprocessing, the input video data is

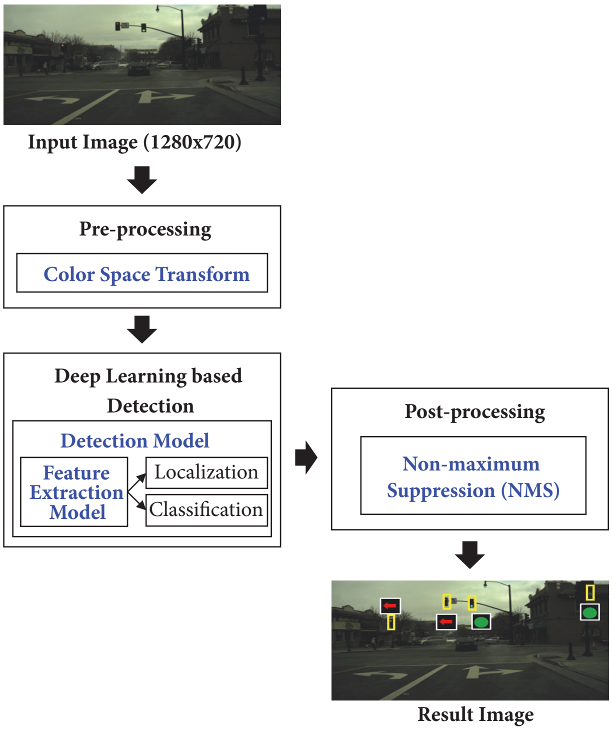


Figure 1: Method overview of deep-learning based traffic light detection.

transformed to other color space. Six color spaces are considered. The deep-learning based detection uses ensemble of feature extraction network model and detection one. Here, we consider three kinds of network models based on the Faster R-CNN and R-FCN which can perform localization and classification. In postprocessing, redundant detection is removed by using the nonmaximum suppression (NMS) technique [56, 57].

We focus on the combination of the color spaces and the ensemble networks which can achieve high performance of the traffic light detection. The color spaces and ensemble network models to be considered are described below.

3.1. Color Spaces.

In the vision-based object detection and classification, it is necessary to determine the color space in which the characteristic of the object appears well. The RGB color space is defined by the three chromatics, red (R), green (G), and blue (B) [58]. For robustness under changes in the lightning condition, normalized RGB has often been used. The normalized RGB denoted as R_n, G_n, and B_n are obtained by R/S, G/S, and B/S, respectively, where $S = R + G + B$ [59]. At low illumination, it is difficult to distinguish between the normalized RGB colors [60]. To overcome this difficulty, Ruta et al. proposed the new red and blue color transform for traffic sign detection [60]. Since the traffic lights are red, green, and yellow, we modify Ruta's color representation for the same. Ruta's red, green, and yellow, denoted as f_R, f_y, and f_G are obtained as below.

$$f_R = \max\left(0, \min\left(R_n - G_n, R_n - B_n\right)\right) \quad (1)$$

$$f_Y = \max\left(0, \min\left(R_n - B_n, G_n - B_n\right)\right) \quad (2)$$

$$f_G = \max\left(0, \min\left(G_n - R_n, G_n - B_n\right)\right) \quad (3)$$

The YCbCr color space is obtained from the RGB [61]. Y component is luma signal, and Cb and Cr are chroma components. The color space can also be represented in cylindrical coordinates such as HSV color space [62]. The hue component, H, refers to the pure color it resembles. All tints, tones, and shades of red have the same hue. The saturation, S, describes how white the color is. The value component, V, also called lightness, describes how dark the color is. The CIE Lab color space consists of one component for luminance, L, and two color components, a and b [63]. It is known that the CIE Lab space is more suitable to many digital image manipulations than RGB color space.

In this paper, the six kinds of color spaces are considered in preprocessing of the traffic light detection system as shown in Figure 1. Each color representation is applied and its performance is compared.

3.2. Deep-Learning Based Ensemble Networks.

It is known that the end-to-end trainable deep-learning models are more efficient than other models in general object detection [35, 36, 44–46], because it allows a sophisticated training by sharing the weights between feature extraction and detection. YOLO [35, 36], Faster R-CNN [44], SSD [45], and R-FCN [46] have been developed for the end-to-end model. In our traffic light detection system, we only consider the end-to-end deep-learning network models that can perform feature extraction and detection.

According to COCO [64], a dataset is divided into three groups depending on the size of the object to be detected; small ($area < 32^2$), medium ($32^2 \leq area \leq 96^2$), and large ($96^2 \leq area$), where area denotes the number of pixels the object occupies. Therefore, the detection performance of a system can be different for different object sizes. Traffic lights are relatively smaller in size than other objects such as vehicle and pedestrian. For example, almost 90 % of traffic lights in our evaluation dataset belong to small-size group ($area < 32^2$) as shown in Table 2. Therefore, it is necessary to determine a deep-learning network model which is suitable for small-size object detection.

Huang et al. applied various network models to general object detection using COCO dataset and their performances are compared [65]. Fourteen kinds of meta-architectures with feature extractors and network models are analyzed. Five feature extractors such as VGGNet [66], MobileNet [67], Inception-v2 [68], Resnet-101 [47], and Inception-Resnet-v2 [69] are compared. Three kinds of network models based on the Faster-RCNN, R-FCN, and SSD are compared. They show that SSD (similar to YOLO) has higher performance for medium and large sized objects, but significantly lower performance than Faster R-CNN and R-FCN for small objects. They show that three ensemble networks such as Faster-RCNN with Inception-Resnet-v2, Faster R-CNN with Resnet-101, and R-FCN with Resnet-101 have higher performances than others for the small-size object detection. This

Table 2: The number (%) of small, medium, and large sizes traffic lights dataset.

Types	# of small (%)	# of medium (%)	# of large (%)	Total
green	7,192 (86.83)	1,091 (13.17)	0 (0.00)	8,283
red	4,694 (95.37)	226 (4.59)	2 (0.04)	4,922
yellow	652 (92.22)	55 (7.78)	0 (0.00)	707
red left	1,429 (82.17)	308 (17.71)	2 (0.12)	1,739
green left	225 (75.50)	62 (20.81)	11 (3.69)	298
off	1,031 (89.42)	122 (10.58)	0 (0.00)	1,153
Total	15,223 (89.01)	1,864 (10.90)	15 (0.09)	17,102

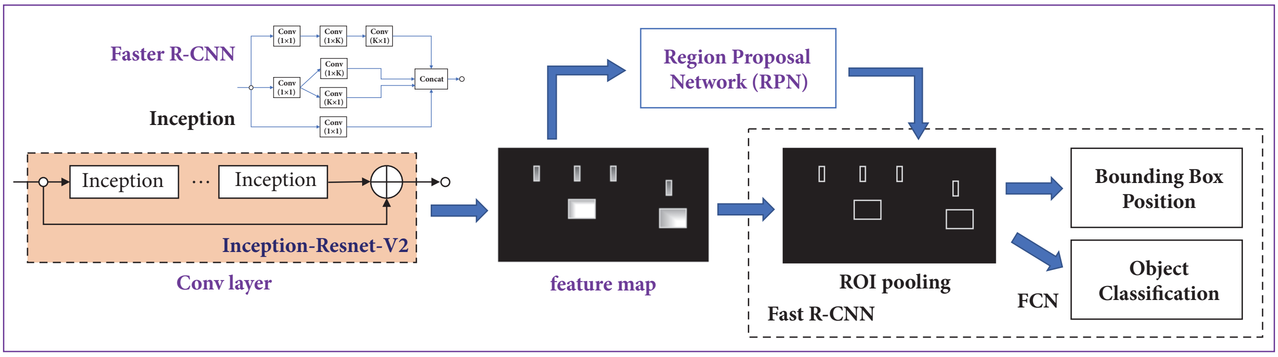


Figure 2: Block diagram of Faster R-CNN network model with Inception-Resnet-v2.

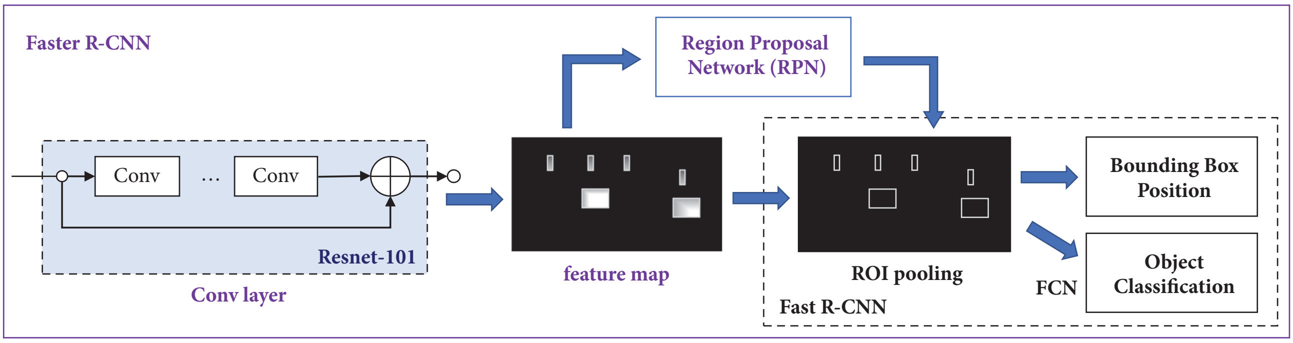


Figure 3: Block diagram of Faster R-CNN network model with Resnet-101.

is the reason why these three ensemble networks are applied in our traffic light detection method.

In Faster R-CNN [44], the selective search is replaced by very small convolutional network called RPN to generate regions of interest (RoI). To handle the variations in aspect ratio and scale of objects, Faster R-CNN introduces the idea of anchor boxes. At each location, three kinds of anchor boxes are used for scale 128×128, 256×256, and 512×512. Similarly, three aspect ratios 1:1, 2:1, and 1:2 are used. RPN predicts the probability of being background or foreground for nine anchor boxes at each location. The remaining network is similar to the Fast-RCNN model. It is known that Faster-RCNN is 10 times faster than Fast-RCNN while maintaining a similar accuracy level [44].

R-FCN [46] is a region-based object detection framework leveraging deep fully convolutional networks. In contrast to other region-based detectors such as Fast R-CNN and Faster R-CNN that apply per-region subnetwork hundreds of times, the region-based detector of R-FCN uses fully convolutional network that applies on the entire image. Instead of RoI pooling at the end layer of Faster R-CNN, R-FCN uses position-sensitive score maps and position-sensitive RoI pooling layer to address a dilemma between translation-invariance in image classification and translation-variance in object detection.

The fully convolutional image classifier backbones, such as Resnet-101 [47] and Inception-Resnet-v2 [69], can be used for object detection. Resnet [47] is a residual learning framework to make the training easy for deeper neural network. It is reported that the residual network with 101 layers (Resnet-101) has the best performance for object classification [47]. Inception-Resnet-v2 [69] is a hybrid inception version which combines residual network and inception network.

In this paper, three kinds of deep-learning based ensemble network models are considered for the traffic light detection (see Figures 2–4). The first network model, Faster-RCNN

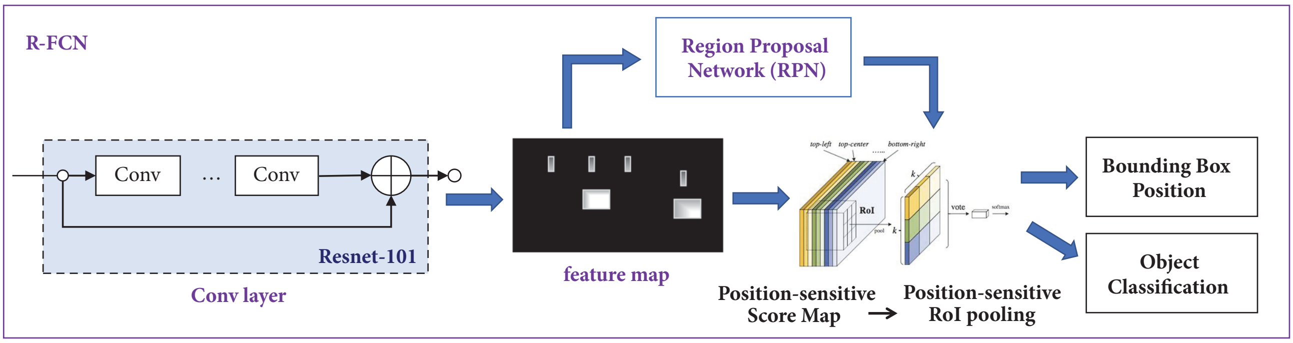


FIGURE 4: Block diagram of R-FCN network model with Resnet-101.

with Inception-Resnet-v2, consists of Inception-Resnetv2 for feature extraction, RPN for candidate extraction, and RoI pooling of Fast R-CNN for classification. The second one, Resnet-101 with Faster R-CNN, consists of Resnet-101 for feature extraction, RPN, and RoI pooling. R-FCN with Resnet-101 consists of Resnet-101, RPN, and position-sensitive score map and position-sensitive RoI pooling for classification. Each network model is applied and its performance is compared.

4. Configuration for Evaluation

In this section, we introduce the dataset and data augmentation method for traffic light detection, parameter tuning, and measurement metrics.

4.1. Dataset and Data Augmentation. For the simulations, we use Bosch Small Traffic Lights Dataset (BSTLD) offered by Behrendt et al. [54]. To use the same types of traffic lights both for training and test, we use only training data set of BSTLD which consists of 5,093 images. Among them, 2,042 images containing 4,306 annotated traffic lights are randomly selected and used as the test data set. The training data set consists of 6,102 images containing 12,796 annotated traffic lights. For testing set, 3,051 images are obtained from BSTLD training set and the others are generated using the following data augmentation techniques.

(i) **Additional Noise and Blur**. Random addition of Gaussian, speckle, salt and pepper noise, and generation of an image with signal-dependent Poisson noise.

(ii) **Brightness Changes in the Lab Space**. Addition of random values to luminance (lightness) component.

(iii) **Saturation and Brightness Changes in the HSV Space**. Additive jitter which is generated at random by means of exponentiation, multiplication and addition of random values to the saturation and value channels.

Both training and testing data sets consist of 1,280×720 size images with annotations including bounding boxes of traffic lights as well as the current state of each traffic light. An active traffic light is annotated by one of six kinds of traffic light states (green, red, yellow, red left, green left, and off) as shown in Figure 5. Detail descriptions of training and testing data sets are summarized in Table 3.

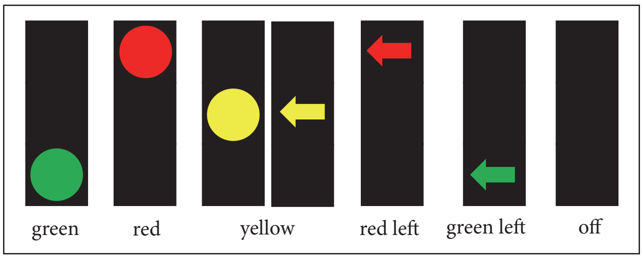


FIGURE 5: Types of traffic lights.

4.2. Parameter Tuning for Training. All three ensemble networks are trained until a maximum of 20,000 epochs using the pretrained weights are obtained from the COCO dataset [64]. The Faster R-CNN and R-FCN networks are trained by stochastic gradient descent (SGD) with momentum [70, 71], where the batch size is 1 and the momentum optimizer value is 0.9. We manually tune the learning rate schedules individually for each feature extractor. In our implementation, the tuning parameters of learning rate for SGD with momentum optimizer are set as follows:

(i) Initial learning rate: 0.0003

(ii) Learning rate of $0 \leq$ Step $< 900{,}000$: 0.0003

(iii) Learning rate of $900{,}000 \leq$ Step $< 1{,}200{,}000$: 0.00003

(iv) Learning rate of $1{,}200{,}000 \leq$ Step: 0.000003

As suggested by Huang et al. [65], we limit the number of proposals to 50 in all three networks to attain similar speeds of traffic light detection.

4.3. Measurement Metrics. To evaluate the performances of traffic light detection, we use measurement metrics such as average precision (AP), mean average precision (mAP), overall AP, and overall mAP that have been widely used in VOC challenge [72, 73] and the COCO 2015 detection challenge [74].

Table 3: Descriptions of training and testing sets.

Dataset	# of images	# of annotated traffic lights	Ratio
Training set	6,102	12,796 (Total classes: 6 ea) (1) green (6,152) (2) red (3,730) (3) yellow (526) (4) red left (1,294) (5) green left (240) (6) off (854)	75 %
Testing set	2,042	4,306 (Total classes: 6 ea) (1) green (2,131) (2) red (1,192) (3) yellow (181) (4) red left (445) (5) green left (58) (6) off (299)	25 %

Table 4: Detection performances (overall mAP and overall AP) of combination methods on test set.

Combination Method		Overall mAP (%)			Overall AP (%)					
Ensemble Network Model	Color Space	total	small	non small	green	red	yellow	red left	green left	off
Faster R-CNN with Inception-Resnet-v2	**RGB**	**20.40**	**15.85**	36.15	**33.46**	**23.81**	4.75	34.69	17.59	8.08
	Normalized RGB	**19.81**	15.16	**38.10**	**32.15**	**22.29**	**6.06**	**38.28**	11.43	**8.65**
	Ruta's RYG	18.07	13.54	33.33	28.58	20.05	2.39	35.30	17.98	4.11
	YCbCr	16.50	12.71	31.31	29.51	15.25	4.67	31.17	14.33	4.07
	HSV	19.70	**15.41**	37.06	29.23	16.91	**6.74**	36.00	**23.54**	5.77
	CIE Lab	17.64	13.31	34.30	26.62	18.27	5.41	34.63	15.82	5.09
Faster R-CNN with Resnet-101	RGB	19.24	14.67	**37.91**	31.21	20.73	3.79	**36.92**	14.34	**8.44**
	Normalized RGB	17.57	13.54	32.86	29.70	18.20	4.87	33.67	11.99	6.98
	Ruta's RYG	14.72	11.21	28.42	26.55	16.62	4.71	26.27	10.14	4.05
	YCbCr	12.36	9.49	25.02	24.03	10.02	2.83	26.36	8.84	2.05
	HSV	15.76	11.11	32.24	25.07	14.77	5.64	23.06	17.99	8.01
	CIE Lab	10.90	7.63	23.73	19.98	13.79	3.67	20.43	5.28	2.28
R-FCN with Resnet-101	RGB	16.63	11.85	37.27	28.47	13.00	4.92	30.19	**18.32**	4.85
	Normalized RGB	14.50	10.95	29.97	23.57	14.41	2.50	27.87	14.59	4.08
	Ruta's RYG	14.21	10.33	26.66	20.89	9.08	3.01	32.75	13.77	5.72
	YCbCr	13.06	9.44	25.05	21.43	10.01	2.50	24.49	14.63	5.28
	HSV	14.66	10.59	29.52	25.40	9.99	3.17	28.39	15.23	5.78
	CIE Lab	12.24	9.06	23.51	14.58	12.43	1.93	27.87	11.80	4.85

AP is precision averaged across all values of recall between 0 and 1. Here, AP is calculated by averaging the interpolated precision over eleven equally spaced interval of recall value $[0, 0.1, 0.2, \ldots 0.9, 1.0]$ [75]. To evaluate the performance for two or more classes, the average of AP, mAP is calculated by averaging APs over every class. We also use overall AP and overall mAP that are obtained by averaging APs and mAPs, respectively, over the IoU=[0.5, 0.55, 0.60, …, 0.90, 0.95], where IoU stands for interval of intersection over union [73].

5. Simulation Results

In this section, we analyze the simulation results and detection examples. For the evaluation, we use measurement metrics such as overall mAP, overall AP, mAP, and AP. For analysis of the detection examples, we apply NMS.

5.1. Simulation Results. Every eighteen methods combined with six different color spaces and three network models are implemented and compared. Tables 4 and 5 show the detection performances where every combination methods are listed in the left columns. In the tables, bold and underlined numbers indicate the top-ranked method, bold for the second ranked and underlined for the third ranked.

The first two network models, Faster R-CNN model with Inception-Resnet-v2 and Faster R-CNN model with Resnet-101, have roughly better performances than R-FCN model with Resnet-101 in terms of mAP. In all three networks,

Table 5: Detection performances (mAP@0.5 and AP@0.5) of combination methods on test set.

Combination Method		mAP@0.5 (%)			AP@0.5 (%)					
Ensemble Network Model	Color Space	total	small	non small	green	red	yellow	red left	green left	off
Faster R-CNN with Inception-Resnet-v2	**RGB**	**38.48**	**31.27**	57.79	**70.56**	**52.12**	8.49	59.11	27.13	13.44
	Normalized RGB	**38.24**	31.42	59.87	**70.43**	**52.09**	10.98	**63.94**	17.39	**14.60**
	Ruta's RYG	35.94	29.16	52.99	65.02	49.77	06.03	57.87	28.76	8.16
	YCbCr	35.55	29.32	51.83	68.68	41.30	9.53	58.91	26.07	8.83
	HSV	35.13	28.82	56.76	58.55	38.50	**12.94**	57.89	**32.88**	10.04
	CIE Lab	32.19	25.45	53.05	54.26	41.84	8.47	55.71	24.00	8.84
Faster R-CNN with Resnet-101	RGB	37.24	30.25	**61.45**	65.23	47.68	6.82	**63.11**	24.37	**16.23**
	Normalized RGB	34.24	28.32	51.82	64.11	43.46	8.20	57.30	19.63	12.72
	Ruta's RYG	31.96	26.14	50.63	61.55	41.21	**13.01**	50.04	18.11	7.85
	YCbCr	26.17	21.44	42.64	56.82	27.16	5.52	47.45	15.57	4.48
	HSV	30.30	22.69	54.00	52.50	34.45	11.02	41.49	27.88	14.44
	CIE Lab	24.71	18.86	41.38	46.99	33.59	6.56	46.18	9.48	5.48
R-FCN with Resnet-101	RGB	34.88	27.33	**62.19**	64.76	36.48	10.07	55.44	**30.93**	11.57
	Normalized RGB	32.16	26.18	52.80	58.86	38.17	5.99	54.03	26.46	9.43
	Ruta's RYG	31.21	24.78	47.20	55.03	30.14	7.13	60.38	23.28	11.29
	YCbCr	30.42	23.18	50.18	57.18	29.10	5.45	49.49	30.42	10.87
	HSV	30.05	23.25	51.61	56.33	28.48	7.58	50.58	26.10	11.25
	CIE Lab	27.33	21.63	44.38	46.86	32.74	4.41	48.95	21.54	9.50

RGB and normalized RGB have high performance than other colors. There is no method having good performance over every type of traffic light.

In Table 4, from the view point of color space, RGB, normalized RGB, and HSV spaces have higher mAP in Faster-RCNN with Inception-Resnet-v2. RGB and normalized RGB have good performance in Faster R-CNN with Resnet-101. In the case of yellow traffic light, normalized RGB and HSV in Faster R-CNN with Inception-Resnet-v2 and HSV in Faster R-CNN model with Resnet-101 have higher performance than other methods, but most methods have limited overall mAPs depending on sizes of traffic light, small and nonsmall. The medium and large size data are combined into nonsmall set, because our dataset has very limited number of large size traffic light data. The top-ranked three methods such as RGB based Faster R-CNN with Inception-Resnetv2, normalized RGB based Faster R-CNN with Inception-Resnet-v2, and HSV based Faster R-CNN with Inception-Resnet-v2 retain their good performances in the small-size object.

Table 5 shows the detection performances in terms of AP and mAP when IoU is fixed to 0.5. The top-ranked two methods such as RGB based Faster R-CNN with Inception-Resnet-v2 and normalized RGB based Faster R-CNN with Inception-Resnet-v2 have also better performance than others, even though the ranking order is slightly changed depending on the object size. Ruta's RYG and HSV in Faster-RCNN model with Resnet-101 and HSV in Faster R-CNN with Inception-Resnet-v2 have significantly high performance for yellow traffic light. Similar to Table 4, CIE Lab color space has relatively poor performance regardless of the network models. As shown in Table 5, the performances of mAP depending on the size are similar to Table 4.

5.2. Detection Examples. After the traffic light detection procedure, we use NMS to remove the redundant detections. At final test process, IoU threshold of NMS is fixed to 0.5. The traffic light detection examples of the top-ranked two methods are shown in Figures 6 and 7. Six example images are selected to show detection results for six types of traffic lights. The traffic lights with object score being greater than 0.5 are detected and classified. True positives are indicated by the corresponding traffic light symbol. False positives and false negatives are noted by FP and FN, respectively. As shown in Figure 6, the top-ranked method, RGB color-based Faster R-CNN with Inception-Resnet-v2, has twenty-six true positives, three false positives, and four false negatives in six images. Figure 7 shows that normalized RGB color-based Faster R-CNN with Inception-Resnet-v2 has twenty-four true positives, two false positives, and seven false negatives. Both methods cannot detect the yellow traffic lights well.

5.3. Summary. Based on the performance analysis, the Faster R-CNN model is more suitable to traffic light detection than R-FCN. Inception-Resnet-v2 shows better performance for feature extraction than Resnet-101 in Faster R-FCN framework. From view point of color space, the use of RGB has highest performance in all ensemble networks. The normalized RGB is also a good color space for Inception-Resnet-v2 model.

6. Conclusions

In this paper, we present a deep-learning based traffic light detection system that consists mainly of color space transform and ensemble network model. Through the simulations,

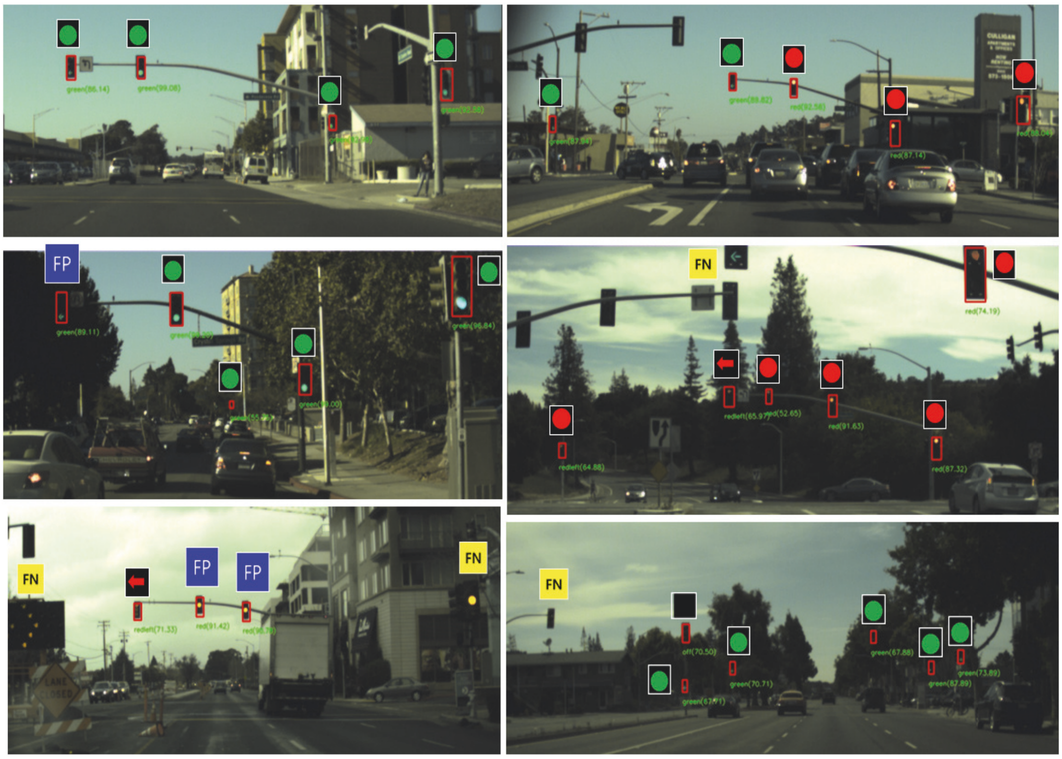


Figure 6: Traffic light detection examples of top-ranked RGB based Faster R-CNN model with Inception-Resnet-v2.

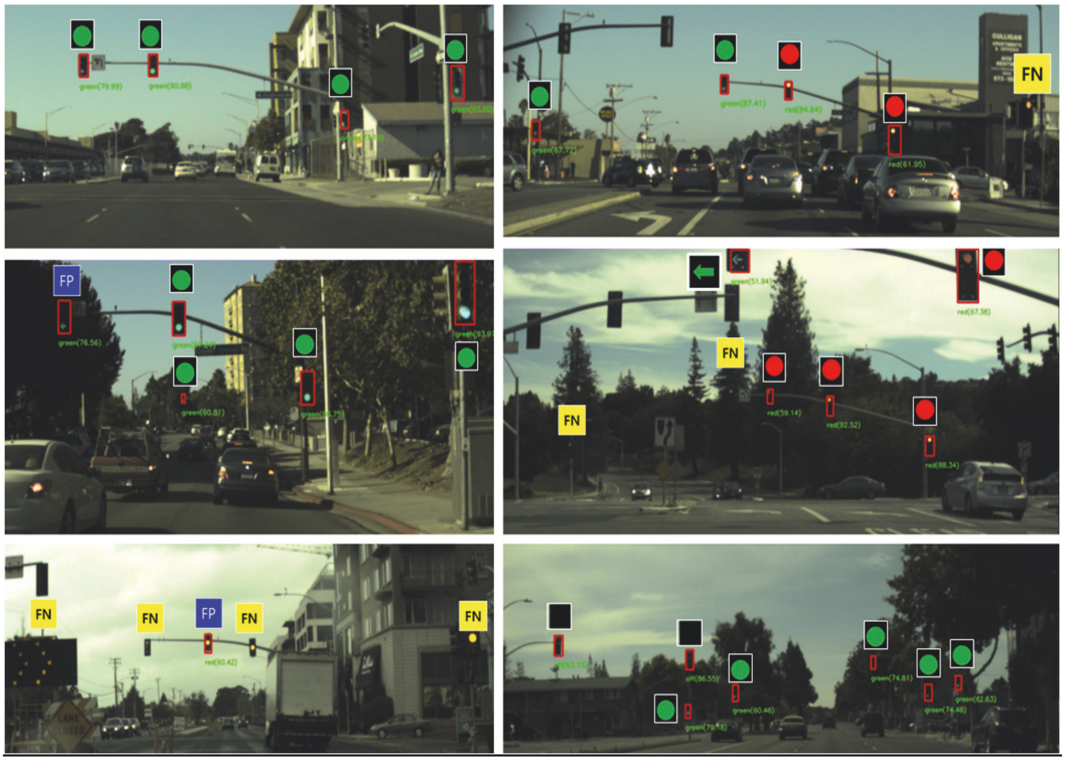


Figure 7: Traffic light detection examples of the second-ranked normalized RGB based Faster R-CNN model with Inception-Resnet-v2.

it is shown that Faster R-CNN with Inception-Resnet-v2 model is more suitable to traffic light detection than others. Regardless of the network models, RGB and normalized RGB color spaces have high performance. limitation However, most methods have limited performance to detect yellow traffic lights. It is observed that yellow lights are often misclassified into red lights because the amount of yellow lights is relatively much smaller in training dataset. The performance can be improved, if yellow light training data is large enough as other colors. The results can help developers to choose appropriate color space and network model when deploying deep-learning based traffic light detection. future research

Data Availability

For the simulations, Bosch Small Traffic Lights Dataset offered by Behrendt et al. is used: https://hci.iwr.uni-heidelberg.de/node/6132.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

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 no. 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 MSIT (Ministry of Science, ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2016-0-00313) supervised by the IITP (Institute for Information & Communications Technology Promotion).

References

[1] S. O.-R. A. V. S. Committee et al., "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems" (Arabic), SAE International, 2014.

[2] R. De Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 333–338, USA, October 2009.

[3] R. De Charette and F. Nashashibi, "Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates," in *Proceedings of the 2009 IEEE Intelligent Vehicles Symposium*, pp. 358–363, China, June 2009.

[4] J. Park and C. Jeong, "Real-time signal light detection," in *Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia (FGCNS)*, pp. 139–142, Hinan, China, December 2008.

[5] C. Yu, C. Huang, and Y. Lang, "Traffic light detection during day and night conditions by a camera," in *Proceedings of the 2010 IEEE 10th International Conference on Signal Processing, ICSP2010*, pp. 821–824, China, October 2010.

[6] M. Omachi and S. Omachi, "Traffic light detection with color and edge information," in *Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009*, pp. 284–287, China, August 2009.

[7] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung, "Night- time traffic light detection based on svm with geometric moment features," *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 7, no. 4, pp. 472–475, 2013.

[8] H.-K. Kim, K. H. Park, and H.-Y. Jung, "Effective traffic lights recognition method for real time driving assistance system in the daytime, World Academy of Science," *Journal of Engineering and Technology*, vol. 59th, 2011.

[9] H.-K. Kim, K. H. Park, and H.-Y. Jung, "Vision based Traffic Light Detection and Recognition Methods for Daytime LED Traffic Light," *Journal of IEMEK*, vol. 9, no. 3, pp. 145–150, 2014.

[10] H. Tae-Hyun, J. In-Hak, and C. Seong-Ik, "Detection of traffic lights for vision-based car navigation system," in *Advances in Image and Video Technology*, vol. 4319 of *Lecture Notes in Computer Science*, pp. 682–691, Springer, Berlin, Germany, 2006.

[11] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation, ICRA 2011*, pp. 5784–5791, China, May 2011.

[12] C. Jang, S. Cho, S. Jeong, J. K. Suhr, H. G. Jung, and M. Sunwoo, "Traffic light recognition exploiting map and localization at every stage," *Expert Systems with Applications*, vol. 88, pp. 290–304, 2017.

[13] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D semantic scene priors for online traffic light interpretation," in *Proceedings of the IEEE Intelligent Vehicles Symposium, IV 2015*, pp. 573–578, Republic of Korea, July 2015.

[14] G. Siogkas, E. Skodras, and E. Dermatas, "Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information," in *Proceedings of the International Conference on Computer Vision Theory and Applications, VISAPP 2012*, pp. 620–627, Italy, February 2012.

[15] M. Omachi and S. Omachi, "Detection of traffic light using structural information," in *Proceedings of the 2010 IEEE 10th International Conference on Signal Processing, ICSP2010*, pp. 809–812, China, October 2010.

[16] M. Diaz-Cabrera, P. Cerri, and P. Medici, "Robust real-time traffic light detection and distance estimation using a single camera," *Expert Systems with Applications*, vol. 42, no. 8, pp. 3911–3923, 2015.

[17] Z. Cai, M. Gu, and Y. Li, "Real-time arrow traffic light recognition system for intelligent vehicle," in *in Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, p. 1, 2012.

[18] M. Michael and M. Schlipsing, "Extending traffic light recognition: Efficient classification of phase and pictogram," in *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Killarney, Ireland, July 2015.

[19] R. M. Haralick, "A measure for circularity of digital figures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 4, no. 4, pp. 394–396, 1974.

[20] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, “Design of an image edge detection filter using the sobel operator,” *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[21] J. Illingworth and J. Kittler, “The adaptive hough transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 690–698, 1987.

[22] T. S. Lee, “Image representation using 2D Gabor wavelets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[23] J. Nishimura and T. Kuroda, “Low cost speech detection using Haar-like filtering for sensornet,” in *Proceedings of the 2008 9th International Conference on Signal Processing, ICSP 2008*, pp. 2608–2611, China, October 2008.

[24] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05)*, vol. 1, pp. 886–893, June 2005.

[25] M. K. Hu, “Visual pattern recognition by moment invariant,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[26] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[27] J. Lewis, “Fast template matching,” *Vision Interface*, vol. 95, pp. 120–123, 1995.

[28] C. Liu and H. Wechsler, “Independent component analysis of Gabor features for face recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 4, pp. 919–928, 2003.

[29] T. Li, S. Zhu, and M. Ogihara, “Using discriminant analysis for multi-class classification: An experimental investigation,” *Knowledge and Information Systems*, vol. 10, no. 4, pp. 453–472, 2006.

[30] P. Cunningham and S. J. Delany, “k-nearest neighbour classifiers,” *Multi- ple Classifier Systems*, vol. 34, pp. 1–17, 2007.

[31] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[32] K.-B. Duan and S. S. Keerthi, “Which is the best multiclass SVM method? An empirical study,” in *Multiple Classifier Systems*, vol. 3541 of *Lecture Notes in Computer Science*, pp. 278–285, Springer, Berlin, Germany, 2005.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS ’12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 779–788, July 2016.

[36] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, Honolulu, HI, July 2017.

[37] A. M. Cailean, B. Cagneau, L. Chassagne, M. Dimian, and V. Popa, “Novel receiver sensor for visible light communications in automotive applications,” *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4632–4639, 2015.

[38] N. Kumar, N. Lourenço, D. Terra, L. N. Alves, and R. L. Aguiar, “Visible light communications in intelligent transportation systems,” in *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, IV 2012*, pp. 748–753, Spain, June 2012.

[39] T. Saito, S. Haruyama, and M. Nakagawa, “A new tracking method using image sensor and photo diode for visible light road-to-vehicle communication,” in *Proceedings of the 2008 10th International Conference on Advanced Communication Technology*, pp. 673–678, Republic of Korea, February 2008.

[40] T. Yamazato, I. Takai, H. Okada et al., “Image-sensor-based visible light communication for automotive applications,” *IEEE Communications Magazine*, vol. 52, no. 7, pp. 88–97, 2014.

[41] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” 2013, https://arxiv.org/abs/1312.6229.

[42] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.

[43] R. Girshick, “Fast r-cnn,” arXiv:1504.080832015.

[44] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[45] W. Liu, D. Anguelov, D. Erhan et al., “SSD: single shot multibox detector,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9905, pp. 21–37, 2016.

[46] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in *Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016*, pp. 379–387, Spain, December 2016.

[47] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, July 2016.

[48] F. Yang, W. Choi, and Y. Lin, “Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2129–2137, Las Vegas, Nev, USA, July 2016.

[49] J. Ren, X. Chen, J. Liu et al., “Accurate single stage detector using recurrent rolling convolution,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 752–760, USA, July 2017.

[50] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, “Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1827–1836, USA, July 2017.

[51] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6526–6534, Honolulu, HI, July 2017.

[52] S. Saini, S. Nikhil, K. R. Konda, H. S. Bharadwaj, and N. Ganeshan, “An efficient vision-based traffic light detection and state recognition for autonomous vehicles,” in *Proceedings of the 28th IEEE Intelligent Vehicles Symposium, IV 2017*, pp. 606–611, USA, June 2017.

[53] G.-G. Lee and B. K. Park, "Traffic light recognition using deep neural networks," in *Proceedings of the 2017 IEEE International Conference on Consumer Electronics, ICCE 2017*, pp. 277-278, USA, January 2017.

[54] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation, ICRA 2017*, pp. 1370–1377, Singapore, June 2017.

[55] M. B. Jensen, K. Nasrollahi, and T. B. Moeslund, "Evaluating state-of-the-art object detector on challenging traffic light data," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2017*, pp. 882–888, USA, July 2017.

[56] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[57] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Computer Vision – ACCV 2014*, vol. 9003 of *Lecture Notes in Computer Science*, pp. 290–306, Springer International Publishing, Cham, 2015.

[58] S. Süsstrunk, R. Buckley, and S. Swen, "Standard RGB color spaces," in *Proceedings of the Final Program and Proceedings of the 7th IS and T/SID Color Imaging Conference: Color Science, Systems and Applications*, pp. 127–134, USA, November 1999.

[59] W. Wintringham, "Color television and colorimetry," *Proceedings of the IRE*, vol. 39, no. 10, pp. 1135–1172, 1951.

[60] A. Ruta, F. Porikli, S. Watanabe, and Y. Li, "In-vehicle camera traffic sign detection and recognition," *Machine Vision and Applications*, vol. 22, no. 2, pp. 359–375, 2011.

[61] T. Acharya, "Median computation-based integrated color interpolation and color space conversion methodology from 8-bit bayer pattern rgb color space to 12-bit ycrcb color space," US Patent 6,356,276, 2002.

[62] R. G. Kuehni, *Color Space and Its Divisions: Color Order from Antiquity to the Present*, John Wiley Sons, 2003.

[63] C. Connolly and T. Fleiss, "A study of efficiency and accuracy in the transformation from RGB to CIE Lab color space," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 1046–1048, 1997.

[64] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: Common objects in context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8693, no. 5, pp. 740–755, 2014.

[65] J. Huang, V. Rathod, C. Sun et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 3296–3305, USA, July 2017.

[66] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. Lecun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016*, pp. 4955–4959, China, March 2016.

[67] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: Efficient convo- lutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[68] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2818–2826, July 2016.

[69] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI '17)*, pp. 4278–4284, February 2017.

[70] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[71] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pp. 2176–2184, USA, June 2013.

[72] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

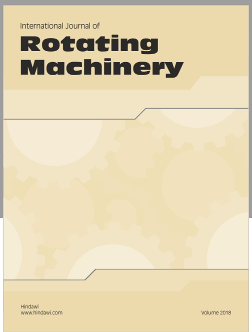
[73] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[74] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

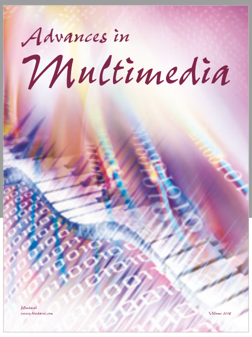
[75] G. G. Chowdhury, *Introduction to Modern Information Retrieval*, Facet publishing, 2010.

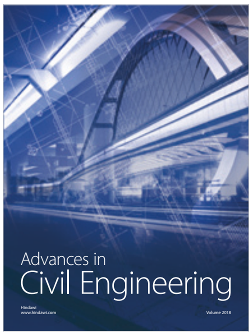
Journal of
Engineering

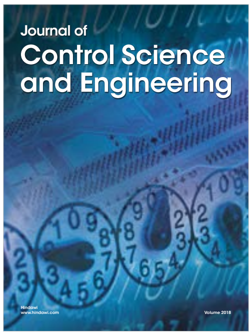
The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

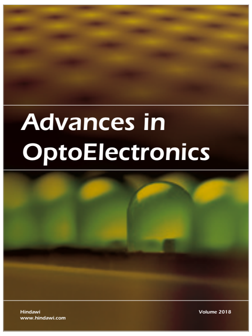

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

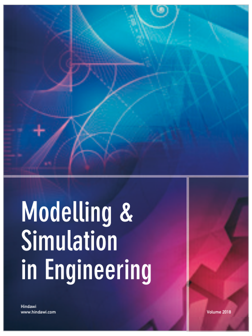
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

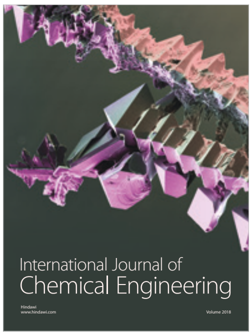

Advances in
OptoElectronics

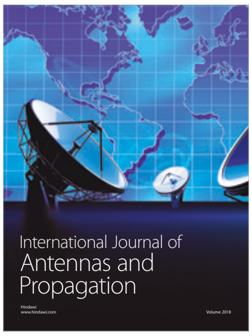
VLSI Design

International Journal of
Navigation and
Observation

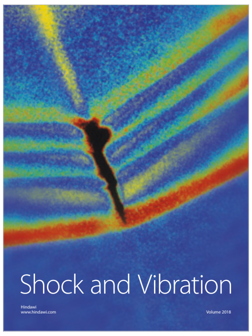

Modelling &
Simulation
in Engineering

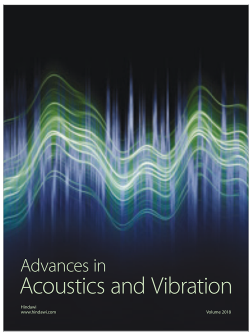
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration