Expert Systems With Applications 48 (2016) 67–75

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

On circular traffic sign detection and recognition

Selcan Kaplan Berkaya*, Huseyin Gunduz, Ozgur Ozsen, Cuneyt Akinlar, Serkan Gunal

Department of Computer Engineering, Anadolu University, Eskisehir, Turkiye

ARTICLE INFO

Keywords:
Circular traffic sign detection
EDCircles
Traffic sign recognition
HOG
Gabor filter
LBP

ABSTRACT

Automatic traffic sign detection and recognition play crucial roles in several expert systems such as driver assistance and autonomous driving systems. In this work, novel approaches for circular traffic sign detection and recognition on color images are proposed. In traffic sign detection, a new approach, which utilizes a recently developed circle detection algorithm and an RGB-based color thresholding technique, is proposed. In traffic sign recognition, an ensemble of features including histogram of oriented gradients, local binary patterns and Gabor features are employed within a support vector machine classification framework. Performances of the proposed detection and recognition approaches are evaluated on German Traffic Sign Detection and Recognition Benchmark datasets, respectively. The results of the experimental work reveal that both approaches offer comparable or even better performances with respect to the best ones reported in the literature and are compatible to real-time operation as well.

1. Introduction

Detection and recognition of traffic signs in digital images have been an important problem for researchers over the last decade. Application areas include advanced driver assistance systems (Timofte, Prisacariu, Van Gool, & Reid, 2011), autonomous driving (Levinson et al., 2011), building and maintaining maps of signs, etc. The problem consists of two successive steps: Traffic Sign Detection (TSD) and Traffic Sign Recognition (TSR). TSD deals with identifying the regions of interest (ROI) and the boundaries of the traffic signs in a given image. A good TSD algorithm must find all relevant traffic signs in an image while producing as few false detections as possible. TSR deals with the classification of a given image patch. A good TSR correctly classifies a given image patch within a pre-formed set of traffic sign classes while making as few false recognitions as possible.

In this paper, we propose methods for circular traffic sign detection and recognition on color images. For detection, we employ the high-speed, parameter-free circle detection algorithm, EDCircles (Akinlar & Topal, 2013), and propose color thresholding algorithms to eliminate false detections. We then quantitatively evaluate the performance of the proposed detection algorithm on the widely-used German Traffic Sign Detection Benchmark (GTSDB) (Houben, Stallkamp, Salmen, Schlipsing, & Igel, 2013). We conclude that the proposed algorithm detects most of the circular traffic signs in the test images. For traffic sign recognition, we try different feature extraction methods ranging from Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Gabor Filter, and a combination of these, and use the Support Vector Machine (SVM) for classification. Finally, we quantitatively evaluate the performance of the proposed recognition methods using the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp, Schlipsing, Salmen, & Igel, 2011).

The rest of this paper is organized as follows: Section 2 introduces the dataset used for quantitative evaluation of the algorithms. In Section 3, we review the related work on traffic sign detection algorithms, present our TSD algorithm, and perform a quantitative evaluation of the proposed method within the GTSDB testbed. In Section 4, we review the literature on TSR, present techniques for TSR of our own, and quantitatively evaluate the proposed methods within the GTSRB testbed. Finally, we conclude the paper in Section 5.

2. Traffic sign dataset

To quantitatively evaluate the performance of TSD and TSR algorithms and to compare and contrast their performance with those of the algorithms presented in the literature, we make use of the widely-used German Traffic Sign Benchmark datasets. Fig. 1 shows the 43 classes of traffic signs in those datasets organized in different subsets. GTSDB is used to evaluate the performance of traffic sign detection algorithms whereas GTSRB is used to evaluate the performance of traffic sign recognition algorithms.

GTSDB consists of 600 training and 300 test images each of size 1360×800 pixels. Each image contains zero or more traffic signs of different colors and shape (Houben et al., 2013). GTSRB, on the other hand, consists of over 50,000 annotations from all of 43 classes of traffic signs (Stallkamp et al., 2011).

* Corresponding author. Tel.: +902223213550; fax: +902223239501.
E-mail addresses: selcankaplan1@gmail.com, selcankaplan@anadolu.edu.tr (S. Kaplan Berkaya), huseyingunduz@anadolu.edu.tr (H. Gunduz), oozsen@anadolu.edu.tr (O. Ozsen), cakinlar@anadolu.edu.tr (C. Akinlar), serkangunal@anadolu.edu.tr (S. Gunal).

http://dx.doi.org/10.1016/j.eswa.2015.11.018
0957-4174/

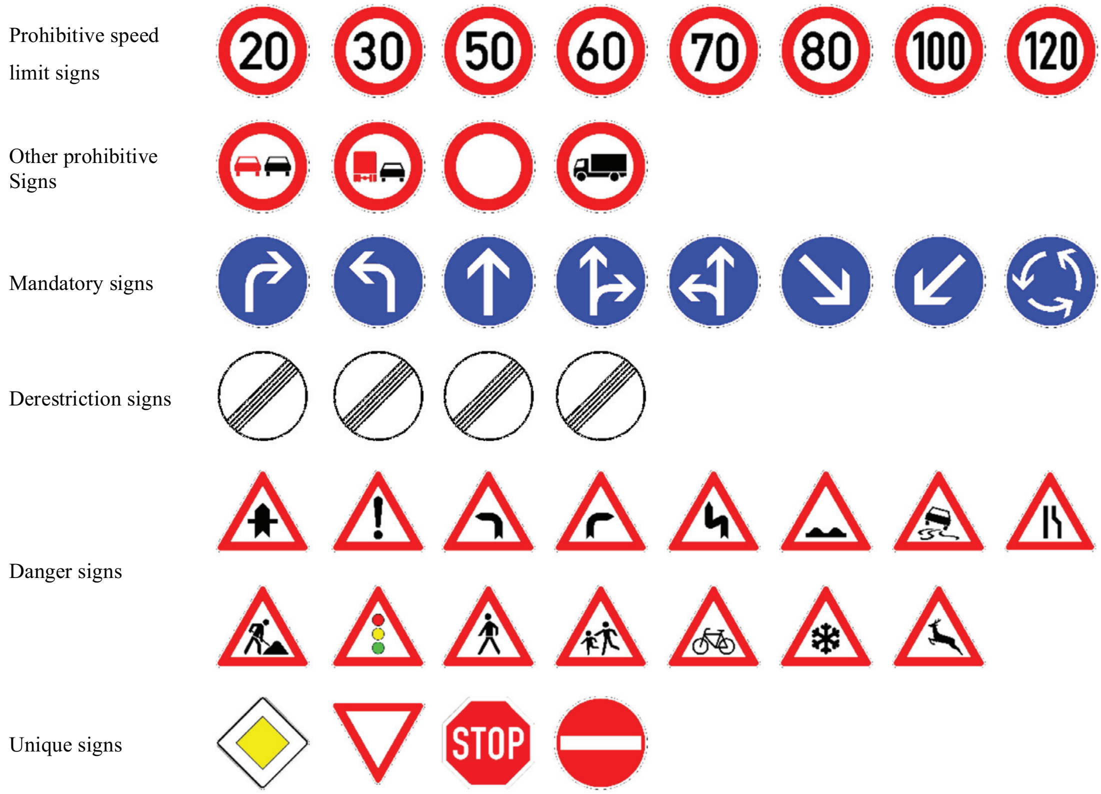

Fig. 1. 43 classes of German traffic sign benchmark datasets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

3. Traffic sign detection

TSD in digital images is traditionally divided into two categories: color-based methods and shape-based methods (Mogelmose, Trivedi, & Moeslund, 2012). Color based detection methods aim to segment a given color image in order to provide a ROI for further processing. The biggest problem of such methods is the difficulty to correctly access the color information in an image due to light intensity variations and illumination changes due to day-night variations and weather conditions (rain, fog, snow, etc.). With color based methods, researchers choose different color spaces and thresholds, and eliminate what they consider to be non-traffic signs. HSI color space, which is less affected from illumination changes and different weather conditions, is very commonly used for segmentation (Maldonado-Bascon, Lafuente-Arroyo, Gil-Jimenez, Gomez-Moreno, & Lopez-Ferreras, 2007; Xu, 2009; Zhu, Zhang, & Lu, 2005). delaEscalera, Moreno, Salichs and Armingol (1997) uses the normalized RGB color space with fixed thresholds in which the red component was chosen as a reference. Authors in (Yalic & Can, 2011) also use RGB color space. Researchers in (Ruta, Li, & Liu, 2010a; Zaklouta & Stanciulescu, 2014) use color enhancement in RGB space. CIELab and Gabor filters are used by Khan, Bhuiyan and Adhami (2011) to represent a color image because this space can independently control color and intensity information. CIECAM97 color model is used in (Gao, Podladchikova, Shaposhnikov, Hong, & Shevtsova, 2006).

In shape-based TSD methods, the usage of Hough Transform (HT) is quite common (Garcia-Garrido, Sotelo, & Martin-Gorostiza, 2006; Loy & Barnes, 2004). Although the methods using HT may offer satisfactory performance, their main disadvantages are high computational complexity and large storage requirements. Some shape-based methods make use of the corners in the image. Distance Transform (DT) is one of these methods. In this method, the corners are found first. DT feature vector image is then obtained by computing the distance of each pixel to the nearest corner (Maldonado-Bascon et al., 2007; Moomivand & Abolfazli, 2011). Although this method is helpful in detecting certain shapes, it takes a long time to create the feature vector, and therefore, it is not suitable for real-time applications (Gavrila, 1999). In (Overett & Petersson, 2011; Xie, Liu, Li, & Qu, 2009), HOG features are also used for the detection of the shapes of traffic signs.

As mentioned above, the main cues for traffic sign detection are color and shape (Salti, Petrelli, Tombari, Fioraio, & Di Stefano, 2015). In order to achieve better results, recent researches combine the color based method with the shape based method (Li, Sun, Liu, & Wang, 2015). Recently, authors in (Li et al., 2015; Lillo-Castellano, Mora-Jimenez, Figuera-Pozuelo, & Rojo-Alvarez, 2015) take both into consideration. In (Lillo-Castellano et al., 2015), color segmentation is based on $L*a*b*$ and HSI spaces, after that machine learning techniques are applied to the traffic sign shape detection.

Some recent works (Greenhalgh & Mirmehdi, 2012; Salti et al., 2015; Yuan, Hao, Chen, & Wei, 2014) have focused on the local stability of traffic sign regions. They use a novel application of maximally stable extremal regions (MSERs) for traffic sign detection, which is reported to be robust to variations in contrast and lighting conditions. The algorithm detects candidates based on the background color instead of border colors of the sign. Then, HOG is adopted to represent the shape features of ROIs and SVM is used to identify traffic signs (Li et al., 2015; Salti et al., 2015).

In (Liu, Chang, & Chen, 2014; Yuan, Guo, Hao, & Chen, 2015), approaches are proposed to handle the difficulties of color and shape based methods. In (Yuan et al., 2015), the authors propose graph based ranking and segmentation algorithms. In (Liu et al., 2014), the authors detect traffic signs without any color and shape information. However, computational complexity and the number of false alarms increase due to the usage of local features.

3.1. Circular traffic sign detection by EDCircles and color thresholding

In this paper, we propose detecting circular traffic signs by the recently-proposed high-speed, parameter-free circle detection algorithm EDCircles (Akinlar & Topal, 2013). This method has been previously used for TSR in (Gunduz, Kaplan, Gunal, & Akinlar, 2013).

```
// I: Input grayscale image
EDCircles(I)
  1. EdgeSegments = DetectEdgeSegmentsBy_EDPF(I);
  2. Arcs = ExtractArcs(EdgeSegments);
  3. CandidateCircles = CombineArcsToCircles(Arcs);
  4. ValidCircles = ValidateCircles(CandidateCircles);
  5. return ValidCircles;
End-EDCircles
```

As seen from the pseudocode given above, EDCircles takes a grayscale image as input and starts by running the parameter-free edge detection algorithm, Edge Drawing Parameter Free (EDPF) (Akinlar & Topal, 2012) to detect a set of edge segments, each of which is a linear pixel chain (step 1). Next, circular arcs are extracted from the edge segments (step 2), and the arcs are combined together to generate a set of candidate circles (step 3). This is achieved by grouping arcs having similar radius and close center coordinates together. The candidate circles are finally validated by the Helmholtz principle (Desolneux, Moisan, & Morel, 2004, 2008) to output a set of perceptually valid circles (steps 4 and 5).

The Helmholtz principle is a statistical validation methodology that can be used to validate a predefined set of Gestalts such as edge segments, line segments, arcs, circles, ellipses, etc. It is an ‘a contrario’ validation method based on the computational Gestalt theory, where the objects are detected as outliers of a suitable background model. Desolneux et al. show that a suitable background model is the Gaussian white noise image, in which all pixels, thus their gradient orientations, are independent. According to the theory, no detections should be made within the Gaussian white noise image, and any large deviation from the background model is considered to be perceptually visible if it corresponds to a predefined Gestalt; a circle in our case. More details on circle validation by the Helmholtz principle and its mathematical formulation can be found in the original EDCircles paper (Akinlar & Topal, 2013).

As stated above, EDCircles works with grayscale images. Since we are working with color images for circular traffic sign detection, we can make use of the color information to enhance the performance. There are two ways to utilize the color information: (1) Firstly, run EDCircles on the grayscale version of the image to detect a set of circles. The detected circles would be boundaries of both some real circular traffic signs (true positives), and some other circular non-traffic sign objects (false positives). We then apply color thresholding for the pixels of the detected circles to eliminate false detections. We call this method EDCircles + Color Thresholding. (2) In the second approach, the color thresholding is applied first to get a binary image, which is then fed into EDCircles to detect the circular boundaries of the white objects in the thresholded image. We call this method Color Thresholding + EDCircles.

Color thresholding is a common technique in traffic sign detection and there are different methods. Since most traffic signs contain red or blue colors, thresholding techniques concentrate mostly on these two colors. One technique is to make use of Normalized RGB (RGBN) color space, presented in (Gomez-Moreno, Maldonado-Bascon, Gil-Jimenez, & Lafuente-Arroyo, 2010). This space is used to decrease the effects of illumination changes and allows finding the correct thresholds. The mask equations for red and blue colors are the following (Gomez-Moreno et al., 2010):

$$\mathrm{Red}(i,j) = \begin{cases} True, \ if r(i,j) \geq ThR \text{ and } g(i,j) \leq ThG \\ False, \ otherwise \end{cases}$$
$$Blue(i,j) = \begin{cases} True, \ if b(i,j) \geq ThB \\ False, \ otherwise \end{cases} \quad (1)$$

Table 1
Threshold values for RGBN, RGBNDiff and the proposed method.

Method	Threshold values
RGBN	$ThR = 0.4, ThG = 0.3, ThB = 0.4$
RGBNDiff	$ThA = 0.17, ThW = 180, ThL = 60$
Proposed	$ThR_1 = 2.5, ThB_1 = 0.65$

RGBN Differences (RGBNDiff) is another method explored in (Gomez-Moreno et al., 2010). This space is especially useful to detect traffic signs that contain black and white colors. The idea is to find pixels with no color information, i.e., gray pixels, and pass them through while suppressing pixels that have color information. This is called chromatic/achromatic decomposition. The formulas used for RGBNDiff are the following (Gomez-Moreno et al., 2010):

$$Achr(i,j) = \begin{cases} True, \ if |r(i,j) - g(i,j)| \leq ThA \\ \quad and \ |r(i,j) - b(i,j)| \leq ThA \\ False, \ otherwise \end{cases} \quad (2)$$

$$White(i,j) = \begin{cases} True, \ if Achr(i,j) = True \\ \quad and \ (R+G+B) \geq ThW \\ False, \ otherwise \end{cases}$$
$$Black(i,j) = \begin{cases} True, \ if (R+G+B) \leq ThL \\ False, \ otherwise \end{cases} \quad (3)$$

In this paper, we propose using the RGB color space but apply the following equations for red and blue color thresholding:

$$\mathrm{Red}(i,j) = \begin{cases} True, \ if r(i,j) \geq g(i,j) \ and \ r(i,j) \geq b(i,j) \\ \quad and \ g(i,j)/|r(i,j) - g(i,j)| \leq ThR_1 \\ False, \ otherwise \end{cases} \quad (4)$$

$$Blue(i,j) = \begin{cases} True, \ if b(i,j) \geq r(i,j) \ and \ g(i,j)/b(i,j) \leq ThB_1 \\ False, \ otherwise \end{cases} \quad (5)$$

The threshold values used in this work are listed in Table 1.

Fig. 2 shows three color images, each of size 1360 × 800, from GTSDB and the resulting binary images after the proposed red thresholding algorithm is applied. Clearly, the red parts of the image have been preserved and all other parts of the image have been wiped out from further consideration.

Fig. 3 shows another set of three sample images from the GTSDB and the resulting binary images after the proposed blue thresholding algorithm is applied. It should be clear from the given examples that all blue areas containing traffic signs have been preserved while all irrelevant regions have been eliminated from further consideration.

Yet another approach is to make use of alternative color spaces instead of RGB. HSV is one such color space, HSI and CIELab are the others. In our experiments we conclude that HSV color space is better for the detection of blue traffic signs compared to RGBN, but it does not produce good results for the detection of red traffic signs. We conclude that the best performing color space is the RGB with the proposed thresholding algorithms.

3.2. TSD experiments

To quantitatively evaluate the performance of the proposed TSD algorithm, we make use of GTSDB presented in Section 2. Of the 43 classes of traffic signs in GTSDB shown in Fig. 1, 20 circular traffic sign classes are the subject of this paper, which are grouped in two different categories as follows: (1) **Prohibitive** signs, which have white background with red border, (2) **Mandatory** signs, which have blue background.

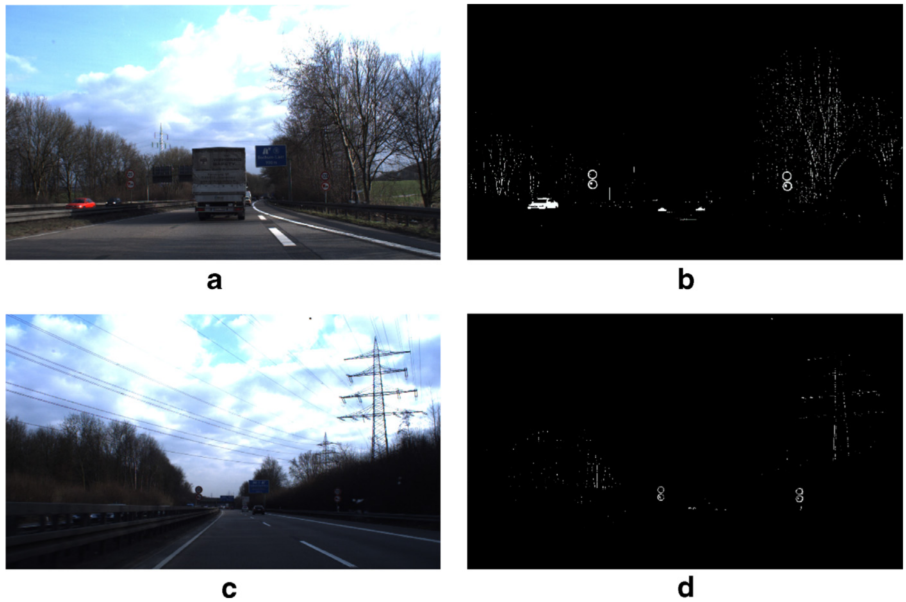

Fig. 2. Color images (a, c) and the resulting binary images (b, d) after the proposed red thresholding. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

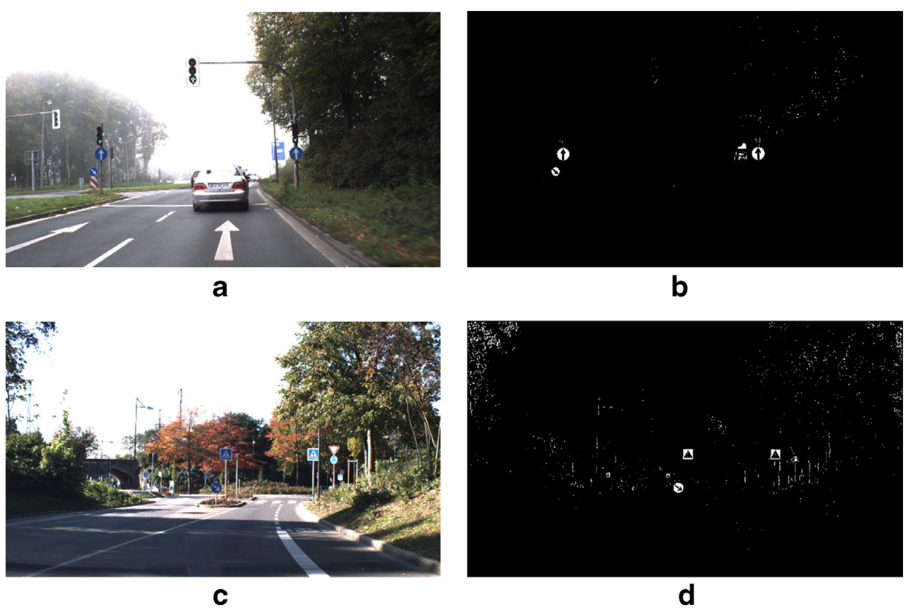

Fig. 3. Color images (a, c) and the resulting binary images (b, d) after the proposed blue thresholding. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Table 2
Number of circular traffic signs from each group in training and test images.

GTSDB	Training images	Test images
Prohibitive	396	161
Mandatory	115	49
Total	591	236

Table 3
The performance of the proposed TSD algorithm on the GTSDB test images.

	Prohibitive			Mandatory		
	TP	FP	F-score	TP	FP	F-score
EDCircles + RGBN	148	130	0.67	32	14	0.67
EDCircles + RGBN + RGBNDiff	137	39	0.80	32	13	0.67
HSV + EDCircles	123	11	0.76	38	6	0.81
RGBN + EDCircles	140	20	0.84	30	9	0.69
RGB + EDCircles + RGBNDiff	151	16	**0.89**	37	1	**0.83**

Table 2 gives a dissection of the number of circular traffic signs from each group within the training and test images of GTSDB. As seen from the table, most of the circular traffic signs are from the prohibitive signs group.

The performances of different variants of the proposed TSD algorithm on the GTSDB test images are presented in Table 3. Within the 300 test images, there are 161 prohibitive and 49 mandatory circular traffic signs. We run two different experiments. In the first set of experiments, we run EDCircles to detect all possible circular shapes and then perform color thresholding to eliminate non-traffic signs. We experimented with two different color thresholding; namely, RGBN and RGBNDiff described in Section 3.1 (Gomez-Moreno et al., 2010).

Table 4
Average processing time (ms) of TSD for (1360 × 800) image.

Method	Part1	Part2	Part3	Total
EDCircles + RGBN	81.10	1.17	-	82.27
EDCircles + RGBN + RGBNDiff	81.10	1.17	~0.01	82.28
HSV + EDCircles	21.16	30.88	-	52.04
RGBN + EDCircles	14.44	25.88	-	40.32
RGB + EDCircles + RGBNDiff	11.69	23.44	1.17	36.30

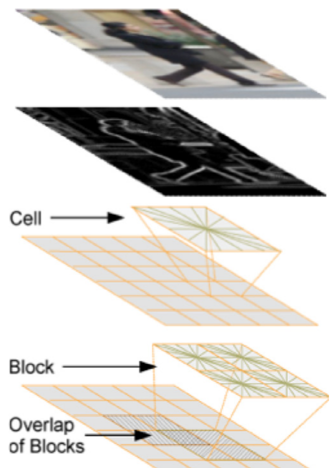

Fig. 4. Structure of HOG features (Dalal & Triggs, 2005).

As seen from the first two rows in Table 3, the performance of this algorithm is not good as it produces many false positives. In the second set of experiments (rows 3–5), we perform color thresholding first to get a binary image, and then run EDCircles. For color thresholding we use HSV, RGBN and our proposed RGB thresholding techniques. As seen from the table, the best results are obtained with our proposed RGB thresholding technique, namely RGB + EDCircles + RGBNDiff.

Considering the computational time of the abovementioned approaches, average processing times of TSD for each approach is measured and provided in Table 4. Those values are obtained with a PC equipped with Intel Core i7-3770 K 3.5 GHz CPU and 16 GB of RAM.

Table 5 compares the performance of the proposed TSD algorithm with the other detection approaches reported in the literature. While the proposed algorithm surpass baseline approaches such as Viola-Jones, it also offers a comparable performance to the best performing ones. Moreover, the proposed TSD algorithm is parameter-free; that is, it runs with a fixed set of parameters for all input images as EDCircles is a parameter-free circle detection algorithm. Therefore, our algorithm does not need any training images to adjust any parameters before it is run for the test images. This is an important property of the proposed algorithm unlike most of the other detection methods.

4. Traffic sign recognition

TSR methods found in the literature are traditionally divided into two categories: template-based techniques and classifier-based techniques.

Template-based approaches involve pixel-based cross-correlation template matching (de la Escalera, Armingol, Pastor, & Rodriguez, 2004; Piccioli, DeMicheli, Parodi, & Campani, 1996). This technique is simple, yet very useful when the tested image and the template images are well aligned. However, geometrical alignment is usually difficult to achieve by automatic sign detection systems, especially when the image is seen against a cluttered background or when it is affected by geometrical distortion. In such cases, the recognition performance degrades quickly. The authors in (Campbell, Egerstedt, How, & Murray, 2010) use DT template matching to classify circular and triangular signs. Using DTs over edge images increases the recognition performance. Similarly, the authors in (Ruta et al., 2010a) use Color Distance Transform, where a separate DT is computed for each color channel. The classification is performed using a nearest neighbor template matching approach.

Classifier-based approaches are based on machine learning techniques. In these approaches, a feature vector is first extracted from the image to reduce computational complexity. Then, the class label of the feature vector is obtained using a classifier such as SVM (Maldonado-Bascon et al., 2007), neural network (NN) (Prieto & Allen, 2009), fuzzy regression tree frameworks (Ruta, Li, & Liu, 2010b) or a random forest based classification method.

In the feature extraction stage, HOG is one of the most successful approaches (Xie et al., 2009). The authors in (Creusen, Wijnhoven, Herbschleb, & de With, 2010) calculate HOG descriptors on each of the color channels to integrate color information. In (Zaklouta & Stanciulescu, 2011), the authors use different sized HOG features and a random forest based classifier. Recently, one of the novel learning algorithms for single-hidden layer feedforward neural networks, namely extreme learning machine (ELM), has been proposed. Using this novel technique with HOG based features, traffic signs are classified in (Sun, Wang, Lau, Seet, & Wang, 2014).

Ojala first proposed the concept of LBP, which is another popular technique to extract features from any given image (Ojala, Pietikainen, & Maenpaa, 2002). A lot of varieties of LBP, such as dominant LBPs (Liao, Law, & Chung, 2009), LBP variance with global matching (Guo, Zhang, & Zhang, 2010), and joint distribution of local patterns with Gaussian mixtures (Lategahn, Gross, Stehle, & Aach, 2010), are proposed for texture classification. In success of the LBP, multi block LBP is used with SVM for traffic sign classification (Samira, Sanae, Mounir, & Youssef, 2014). However, LBP is unable to provide color information, only focuses on local textures while ignoring the distribution of global shapes, and the rotational invariant version of LBP has a very small size. To overcome this insufficiency, a novel Color Global and Local Oriented Edge Magnitude Pattern is proposed in (Yuan et al., 2014) and SVM classifier is used for the recognition purpose. Alternatively, the authors in (Tang & Huang, 2013) proposes combining LBP and HOG features together to enhance the recognition performance.

Jin, Fu, and Zhang (2014) propose a method based on deep learning, in which using hinge loss stochastic gradient method to train the convolution neural and achieves the high recognition rate. Also, Ciresan, Meier, Masci, and Schmidhuber (2011) used the convolution neural network to learn the discriminative features from pixels.

Hazelhoff, Creusen, van de Wouwa and de With (2012) propose a method based on the Bag of Words approach. In order to emphasize at the inter-sign differences, they extend the approach with a flexible modular codebook.

Douville (2000) uses a NN with normalized Gabor Wavelet Transform features from multi resolution filters. The author reports high recognition rate and real-time processing when classifying traffic signs that have background clutter and underwent geometric transformations.

In this paper, on the other hand, we propose an ensemble of the features including HOG, LBP and Gabor employed within a SVM classification framework. Each feature extraction methodology is briefly explained and the experimental work for the proposed TSR approach is provided in the following subsections.

4.1. HOG features

HOG-based features have first been proposed by Dalal and Triggs (2005) for detection of pedestrians. Afterwards, HOG has drawn intense interest and has been used in many other pattern recognition

Table 5
Performance comparison of the proposed method with other methods (Houben et al., 2013).

Team	Method	Prohibitive	Mandatory
wgy@HIT501	ELL_SVM2	1.00	1.00
visics	boosted_intChn_ratio_scales	1.00	0.93
BolognaCVLab	MSER-1+HOG-2+SVM+Heights+TL	0.99	0.90
Proposed	**RGB + EDCircles + RGBNDiff**	**0.89**	**0.83**
INI-RTCV	Viola-Jones	0.85	0.51
meca	linSVM	0.81	0.56
INI-RTCV	Hough-like Voting Scheme	0.42	0.27

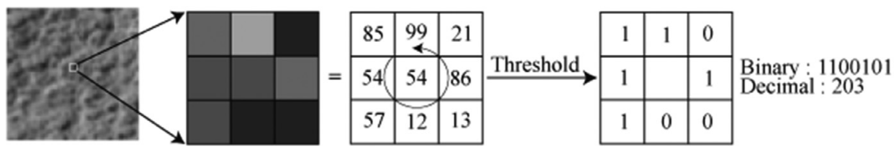

Fig. 5. LBP production steps (Ahonen, Hadid, & Pietikainen, 2006).

problems. TSR is one of those problems. The main idea behind the HOG method is to express an image as a group of local histograms. These histograms consist of both gradient orientations and gradient magnitude.

Fig. 4 shows the structure of HOG. After applying the Sobel operator to the image, gradient orientations and magnitudes are calculated. Image is divided into sub-regions, each called a block. A block slightly overlaps with its neighbors. Each block is further divided into non-overlapping regions called cells. For each cell, histograms are generated by gradient orientation and magnitude. During this process, grouping of orientations and magnitudes are performed for better results. For example, Dalal and Triggs (2005) reports that using nine bins gives the best results. The feature vector for the image is obtained by combining histograms from each cell and the block. In this work, we use 9 bins and a 288-dimensional feature vector for 40 × 40 image patches used for TSR.

4.2. LBP features

Texture analysis in computer vision has a very important place. In the past years, researchers have developed LBP, which is not only simple in terms of calculations, but is also very effective for texture analysis. LBP is not affected by light intensity changes, which is another important property of the method. Computation of LBP at a pixel (x_c, y_c) is performed by making comparisons around the pixel's eight neighbors as in

$$LBP(x_c, y_c) = \sum_{(n=0)}^{7} s(i_n - i_c)2^n, \tag{6}$$

where i_c is the gray level value of the center pixel (x_c, y_c), i_n is gray level value of its 8 neighbors. The function $s(x)$ is defined as follows:

$$s(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}. \tag{7}$$

Shortly stated, for the 8 neighbors, the gray level value of the middle pixel is considered to be a cut-off value. If the gray level value of a neighboring pixel is smaller than this cut-off, the function gives 0; otherwise, it gives 1. Fig. 5 illustrates how LBP is computed for an example pixel. For (3 × 3) window, center pixel consist of 8 neighborhood pixels and a total of $2^8 = 256$ different LBP codes (patterns) are generated, but not all LBP codes are used (Ojala, Pietikainen, & Harwood, 1996). Instead, a pattern is used if 0 to 1 transition within the code is twice or less. Such patterns are called as uniform. For example, 11110011 or 000111111 are uniform patterns. However, 01100110 or 10100110 are non-uniform patterns. In the generation of the LBP histogram, the histogram has a separate bin for every uniform pattern, and all non-uniform patterns are assigned to a single bin. Using uniform patterns, the length of the feature vector reduces from 256 to 59. Uniform LBP is expressed as $LBP^u_{P,R}$, where P is the number of points and R represents the radius. Fig. 6 shows LPB generation with three different (P, R) pairs.

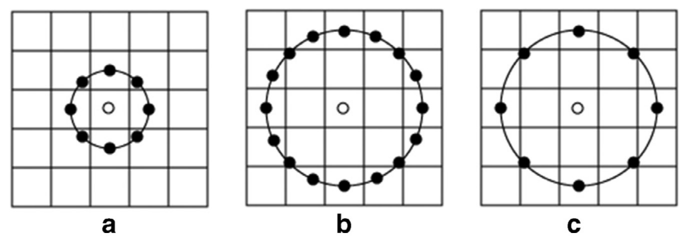

Fig. 6. Original image (a) and LBP operator with different P, R pairs (b, c).

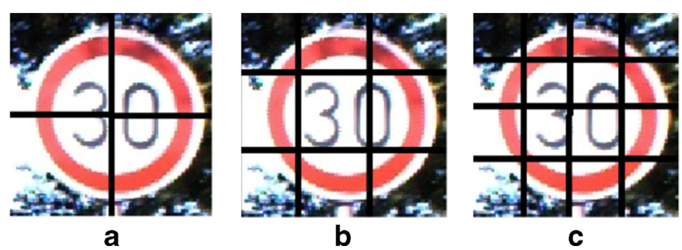

Fig. 7. Image is divided into sub-regions (a) 2 × 2, (b) 3 × 3, (c) 4 × 4.

To compute the feature vector for an image, the image is first divided into a given number of sub-regions as illustrated in Fig. 7. Then $LBP^u_{P,R}$ histogram is obtained for each region; and finally, all histograms are added to one another as shown in Fig. 8. The resulting histogram is used as the final feature vector for the entire image.

In this work, LBP is obtained using 2 × 2 patch size. The radius is set to 1, and 8 neighbors are used. Thus, four 59-bin histograms are generated and combined together. Finally, we get a 236-dimensional feature vector for 40 × 40 images used for TSR.

4.3. Gabor features

Gabor filter is a feature extraction technique commonly used in the field of computer vision and image processing (Jemaa & Khanfir, 2009; Liu & Wechsler, 2002). When the Gabor filter is applied to an image, the pixels having the same local frequency

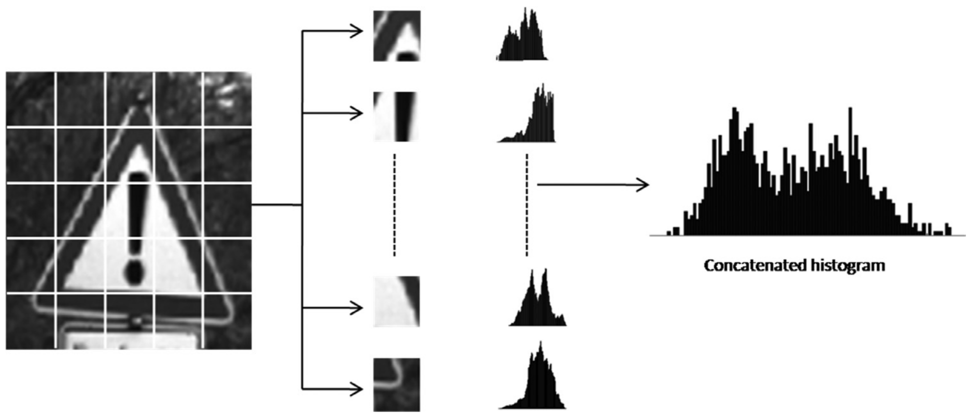

Fig. 8. Combining histograms (Samira et al., 2014).

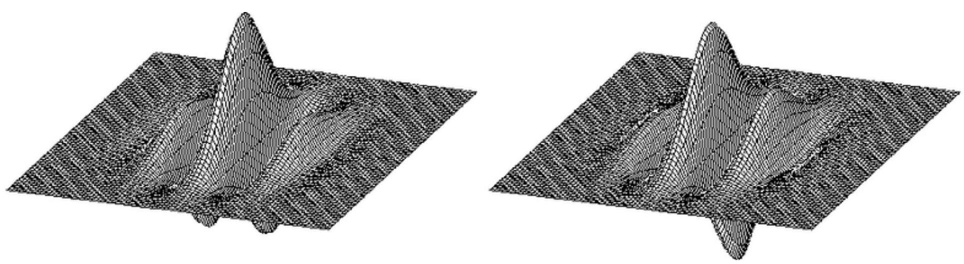

Fig. 9. Real and imaginary parts of a complex Gabor function.

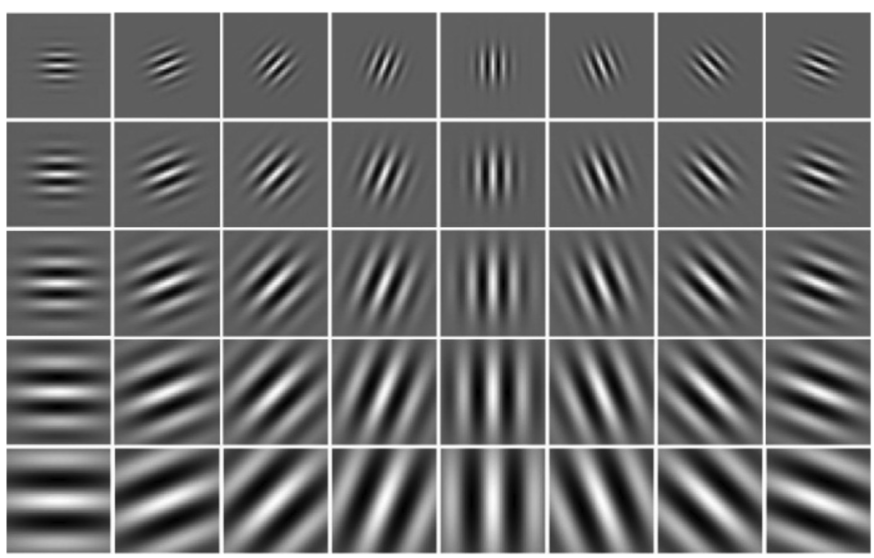

Fig. 10. An example Gabor filter bank.

and orientation will give the strongest response. Gabor filter equation can be expressed mathematically as

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{1}{2}\left(\frac{\tilde{x}^2}{\sigma_x^2} + \frac{\tilde{y}^2}{\sigma_y^2}\right)\right] exp[j2\pi f\tilde{x}] \tag{8}$$

$$\tilde{x} = xcos\theta + ysin\theta \tag{9}$$

$$\tilde{y} = -xsin\theta + ycos\theta, \tag{10}$$

where σ_x and σ_y are the standard deviation of the Gaussian envelope along the x and y axes. θ indicates the orientation of the Gabor filters, and f is a sinusoidal wave frequency. Fig. 9 shows real and imaginary parts of the complex Gabor function. In order to obtain a feature vector, a filter bank containing Gabor filters with different angles and frequencies is used. An example Gabor filter bank is shown in Fig. 10.

In this work, a Gabor filter bank containing 18 filters are used. The filters are generated by 6 different angles with 30 degree increments, i.e., 0, 30, 60, 90, 120, 150, and 3 different octave intervals 4, 8, 16 pixels/cycle. When an input image $I(x, y)$ is processed by a Gabor filter, the output is computed by convolving the image with the corresponding Gabor filter as follows:

$$R_n(x,y) = I_n(x,y) * G_n(x,y). \tag{11}$$

Table 6
Recognition results.

Features	All Signs	Speed Limits	Other Prohibitions	Derestriction	Mandatory	Danger	Unique Signs
LBP	93.36	94.17	98.06	91.94	98.92	85.66	99.85
GABOR	93.90	95.17	98.86	98.33	95.98	91.07	99.26
HOG	94.56	94.58	99.86	90.27	99.32	88.20	100.00
HOG + LBP	95.24	95.34	99.86	91.94	99.43	89.31	100.00
HOG + GABOR	97.00	96.49	99.80	98.88	99.83	94.58	99.95
GABOR + LBP	95.17	95.80	99.46	98.61	98.19	91.36	99.65
GABOR + LBP + HOG	97.04	96.88	99.86	97.50	99.83	94.15	99.95

Table 7
Average feature extraction time for 40×40 image.

Features	Time (ms)
LBP	1.21
GABOR	2.31
HOG	0.02
HOG + LBP	1.24
HOG + GABOR	2.40
GABOR + LBP	3.49
GABOR + LBP + HOG	3.60

4.4. TSR experiments

To quantitatively measure the performance of the proposed TSR algorithm, we make use of GTSRB presented in Section 2. As mentioned before, GTSRB consists of 43 traffic sign classes with over 50,000 annotations. Using the annotation, the image patches are cropped and rescaled to 40×40 pixels before feature extraction. Features are extracted using HOG, LBP and GABOR features individually and in different combinations. While the features are used in combination, the feature vectors are normalized to between 0 and 1, and each one is appended to the other. For classification, SVM is used.

Table 6 presents the obtained classification rates for different feature sets mentioned above. Clearly, the best results are obtained when all feature extractions methods are combined together (GABOR + LBP + HOG) as seen in the last row. Table 7 lists the average processing time for feature extraction for a 40×40 image. It is clear from the table that the feature extraction in the worst case takes less than 4 ms, which is negligible for all practical purposes.

Finally, Table 8 compares the performance of the proposed TSR algorithm with the other TSR algorithms found in the literature. One can note from this table that the proposed algorithm offers comparable or even better performance with respect to the others. While the proposed approach offers the best results in certain categories such as “Other Prohibitions” and “Mandatory”, it also provides very close results to the best ones reported in the other categories.

5. Conclusions

Thanks to the fast growing automotive industry, advanced driver assistance and autonomous driving systems have received increasing attention in recent years. Automatic traffic sign detection and recognition are among the most important and useful features of these expert and intelligent systems. Safe and comfortable drive is guaranteed with the help of these systems. For this reason, novel approaches for circular traffic sign detection and recognition are proposed to further improve the capabilities of automatic traffic sign detection and recognition methods presented in the literature.

While most of the TSD studies in the literature prefer either color-based or shape-based approaches, this work utilizes both approaches together. Similarly, most researchers prefer a single feature set such as LBP and HOG, this work makes use of an ensemble of the most successful features for the TSR problem. In this way, comparable or even better performances are obtained for both approaches with respect to the best results reported in the literature. Besides, it is validated that both approaches are compatible to real-time operation. Surely, the performance of the proposed detection approach depends on the success of circle detection and color thresholding algorithms whereas the profile of the feature sets and the classification algorithm would affect the performance of the proposed recognition algorithm as well. Additionally, hardware resources and the image size have direct impact on the processing time of both approaches; that is, the speed of both approaches can further be increased with more powerful hardware platforms and smaller image sizes.

Analysis of different circle detection algorithms, color models, feature sets and classifiers remain as interesting future studies. Since the proposed approaches are amenable to real-time operation, they may be implemented on appropriate mobile platforms thanks to the increasing processing power and resources of the mobile devices. The proposed approaches may also partly be used in expert and intelligent systems for detection and/or recognition of the objects other than the traffic signs.

Acknowledgment

This work was supported by Anadolu University, Fund of Scientific Research Projects under grant number 1304F061.

Table 8
Comparison of the recognition performances (Stallkamp et al., 2011).

Features	All Signs	Speed Limits	Other Prohibitions	Derestriction	Mandatory	Danger	Unique Signs
cnn hog3	98.98	99.14	99.57	100.00	97.89	98.83	100.00
EBLearn 2LConvNet	98.97	98.87	99.80	99.00	97.78	98.72	100.00
Human Performance	98.81	97.39	99.59	99.67	99.72	99.04	99.90
IKSVM + PHOG + HOG2	97.88	97.91	99.25	99.67	96.78	96.17	99.95
SRC + LDAs I/HOG1/HOG2	97.35	97.63	99.46	100.00	96.05	94.54	99.95
HOG + LDA + VQ	96.87	95.73	98.50	99.33	96.72	95.39	99.90
HOG2 + LDA	96.32	95.76	97.28	99.33	95.00	95.00	99.35
HOG1 + 3-NN	73.89	61.39	87.28	87.00	93.39	53.83	98.76
Proposed	**97.04**	**96.88**	**99.86**	**97.50**	**99.83**	**94.15**	**99.95**

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*, 2037–2041.

Akinlar, C., & Topal, C. (2012). EDPF: A real-time parameter-free edge segment detector with a false detection control. *International Journal of Pattern Recognition and Artificial Intelligence, 26*.

Akinlar, C., & Topal, C. (2013). EDCircles: a real-time circle detector with a false detection control. *Pattern Recognition, 46*, 725–740.

Campbell, M., Egerstedt, M., How, J. P., & Murray, R. M. (2010). Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences, 368*, 4649–4672.

Ciresan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011). A committee of neural networks for traffic sign classification. In *Proceedings of 2011 international joint conference on neural networks* (pp. 1918–1921).

Creusen, I. M., Wijnhoven, R. G. J., Herbschleb, E., & de With, P. H. N. (2010). Color exploitation in HOG-based traffic sign detection. In *Proceedings of 2010 IEEE international conference on image processing* (pp. 2669–2672).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of 2005 IEEE computer society conference on computer vision and pattern recognition: 1* (pp. 886–893).

de la Escalera, A., Armingol, J. M., Pastor, J. M., & Rodriguez, F. J. (2004). Visual sign information extraction and identification by deformable models for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems, 5*, 57–68.

delaEscalera, A., Moreno, L. E., Salichs, M. A., & Armingol, J. M. (1997). Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics, 44*, 848–859.

Desolneux, A., Moisan, L., & Morel, J. M. (2004). Gestalt theory and computer vision. *Thinking and knowing* (pp. 71–101). Kluwer Academic Publishers.

Desolneux, A., Moisan, L., & Morel, J. M. (2008). *From Gestalt theory to image analysis: A probabilistic approach*: 34. Springer.

Douville, P. (2000). Real-time classification of traffic signs. *Real-Time Imaging, 6*, 185–193.

Gao, X. W., Podladchikova, L., Shaposhnikov, D., Hong, K., & Shevtsova, N. (2006). Recognition of traffic signs based on their colour and shape features extracted using human vision models. *Journal of Visual Communication and Image Representation, 17*, 675–685.

Garcia-Garrido, M. A., Sotelo, M. A., & Martin-Gorostiza, E. (2006). Fast traffic sign detection and recognition under changing lighting conditions. In *Proceedings of ITSC* (pp. 811–816).

Gavrila, D. (1999). Traffic sign recognition revisited. *21. DAGM-Symposium* (pp. 86–93). London, UK: Springer- Verlag.

Gomez-Moreno, H., Maldonado-Bascon, S., Gil-Jimenez, P., & Lafuente-Arroyo, S. (2010). Goal evaluation of segmentation algorithms for traffic sign recognition. *IEEE Transactions on Intelligent Transportation Systems, 11*, 917–930.

Greenhalgh, J., & Mirmehdi, M. (2012). Real-time detection and recognition of road traffic signs. *IEEE Transactions on Intelligent Transportation Systems, 13*, 1498–1506.

Gunduz, H., Kaplan, S., Gunal, S., & Akinlar, C. (2013). Circular traffic sign recognition empowered by circle detection algorithm. In *Proceedings of IEEE 21st signal processing and communications applications conference*.

Guo, Z. H., Zhang, L., & Zhang, D. (2010). Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recognition, 43*, 706–719.

Hazelhoff, L., Creusen, I., van de Wouwa, D., & de With, P. H. N. (2012). Large-scale classification of traffic signs under real-world conditions. In *Proceedings of SPIE 8304, multimedia on mobile devices*.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., & Igel, C. (2013). Detection of traffic signs in real-world images: the German traffic sign detection benchmark. In *Proceedings of 2013 international joint conference on neural networks (IJCNN)* (pp. 1–8).

Jemaa, Y. B., & Khanfir, S. (2009). Automatic local Gabor features extraction for face recognition. *International Journal of Computer Science and Information Security, 3*.

Jin, J. Q., Fu, K., & Zhang, C. S. (2014). Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems, 15*, 1991–2000.

Khan, J. F., Bhuiyan, S. M. A., & Adhami, R. R. (2011). Image segmentation and shape analysis for road-sign detection. *IEEE Transactions on Intelligent Transportation Systems, 12*, 83–96.

Lategahn, H., Gross, S., Stehle, T., & Aach, T. (2010). *Texture classification by modeling joint distributions of local patterns with Gaussian mixtures: 19* (pp. 1548–1557).

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V., Sokolsky, M., Stanek, G., Stavens, D., Teichman, A., Werling, M., & Thrun, S. (2011). Towards fully autonomous driving: systems and algorithms. In *Proceedings of 2011 IEEE intelligent vehicles symposium* (pp. 163–168).

Li, H. J., Sun, F. M., Liu, L. J., & Wang, L. (2015). A novel traffic sign detection method via color segmentation and robust shape matching. *Neurocomputing, 169*, 77–88.

Liao, S., Law, M. W. K., & Chung, A. C. S. (2009). *Dominant local binary patterns for texture classification: 18* (pp. 1107–1118).

Lillo-Castellano, J. M., Mora-Jimenez, I., Figuera-Pozuelo, C., & Rojo-Alvarez, J. L. (2015). Traffic sign segmentation and classification using statistical learning methods. *Neurocomputing, 153*, 286–299.

Liu, C. J., & Wechsler, H. (2002). Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing, 11*, 467–476.

Liu, C. S., Chang, F. L., & Chen, Z. X. (2014). Rapid multiclass traffic sign detection in high-resolution images. *IEEE Transactions on Intelligent Transportation Systems, 15*, 2394–2403.

Loy, G., & Barnes, N. (2004). Fast shape-based road sign detection for a driver assistance system. In *Proceedings of intelligent robots and systems: 1* (pp. 70–75).

Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., & Lopez-Ferreras, F. (2007). Road-sign detection and recognition based on support vector machines. *IEEE Transactions on Intelligent Transportation Systems, 8*, 264–278.

Mogelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems, 13*, 1484–1497.

Moomivand, E., & Abolfazli, E. (2011). A modified structural method for shape recognition. In *Proceedings of IEEE symposium on industrial electronics and applications*. Malaysia: Langkawi.

Ojala, T., Pietikainen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition, 29*, 51–59.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 971–987.

Overett, G., & Petersson, L. (2011). Large scale sign detection using HOG feature variants. In *Proceedings of 2011 IEEE intelligent vehicles symposium* (pp. 326–331).

Piccioli, G., DeMicheli, E., Parodi, P., & Campani, M. (1996). Robust method for road sign detection and recognition. *Image and Vision Computing, 14*, 209–223.

Prieto, M. S., & Allen, A. R. (2009). Using self-organising maps in the detection and recognition of road signs. *Image and Vision Computing, 27*, 673–683.

Ruta, A., Li, Y. M., & Liu, X. H. (2010a). Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition, 43*, 416–430.

Ruta, A., Li, Y. M., & Liu, X. H. (2010b). Robust Class Similarity Measure for Traffic Sign Recognition. *IEEE Transactions on Intelligent Transportation Systems, 11*, 846–855.

Salti, S., Petrelli, A., Tombari, F., Fioraio, N., & Di Stefano, L. (2015). Traffic sign detection via interest region extraction. *Pattern Recognition, 48*, 1039–1049.

Samira, E., Sanae, B., Mounir, A. K., & Youssef, F. (2014). Traffic sign recognition based on multi-block LBP features using SVM with normalization. In *Proceedings of 2014 9th international conference on intelligent systems: theories and applications (Sita'14)*.

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The German traffic sign recognition benchmark: a multi-class classification competition. In *Proceedings of 2011 international joint conference on neural networks (IJCNN)* (pp. 1453–1460).

Sun, Z. L., Wang, H., Lau, W. S., Seet, G., & Wang, D. W. (2014). Application of BW-ELM model on traffic sign recognition. *Neurocomputing, 128*, 153–159.

Tang, S. S., & Huang, L. L. (2013). Traffic sign recognition using complementary features. In *Proceedings of 2013 second IAPR Asian conference on pattern recognition* (pp. 210–214).

Timofte, R., Prisacariu, V. A., Van Gool, L. J., & Reid, I. (2011). In C. H. Chen (Ed.), *Combining traffic sign detection with 3d tracking towards better driver assistance*. World Scientific Publishing.

Xie, Y., Liu, L. F., Li, C. H., & Qu, Y. Y. (2009). Unifying visual saliency with HOG feature learning for traffic sign detection. In *Proceedings of 2009 IEEE intelligent vehicles symposium: 1, 2* (pp. 24–29).

Xu, S. (2009). Robust traffic sign shape recognition using geometric matching. *IET Intelligent Transport Systems, 3*, 10–18.

Yalic, H. Y., & Can, A. B. (2011). Automatic recognition of traffic signs in Turkey roads. In *Proceedings of 2011 IEEE 19th signal processing and communications applications conference*.

Yuan, X., Guo, J., Hao, X., & Chen, H. (2015). Traffic sign detection via graph-based ranking and segmentation algorithms. In *Proceedings of IEEE transactions on systems, man and cybernetics: systems* (pp. 1–13).

Yuan, X., Hao, X. L., Chen, H. J., & Wei, X. Y. (2014). Robust traffic sign recognition based on color global and local oriented edge magnitude patterns. *IEEE Transactions on Intelligent Transportation Systems, 15*, 1466–1477.

Zaklouta, F., & Stanciulescu, B. (2011). Warning traffic sign recognition using a HOG-based K-d tree. In *Proceedings of 2011 IEEE Intelligent Vehicles Symposium* (pp. 1019–1024).

Zaklouta, F., & Stanciulescu, B. (2014). Real-time traffic sign recognition in three stages. *Robotics and Autonomous Systems, 62*, 16–24.

Zhu, S. D., Zhang, Y., & Lu, X. F. (2005). Detection for triangle traffic sign based on neural network. In *Proceedings of IEEE international conference on vehicular electronics and safety* (pp. 25–28).