International Journal of Innovative Computing, Information and Control
Volume **14**, Number **1**, February **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 1–14

AN EFFICIENT ALGORITHM FOR IMPLEMENTING TRAFFIC SIGN DETECTION ON LOW COST EMBEDDED SYSTEM

Aryuanto Soetedjo and I Komang Somawirata

Department of Electrical Engineering
National Institute of Technology
Jalan Raya Karanglo KM 2 Malang 65153, Indonesia
aryuanto@gmail.com; kmgsomawirata@yahoo.com

Received May 2017; revised September 2017

Abstract. *This paper presents the implementation of traffic sign detection on a low cost embedded system. The proposed system consists of color thresholding, shape detection, and sign validation. An efficient color thresholding based on the red-blue angle color transformation (RBAT) and the normalized red color is employed to extract the red circular traffic sign in an image. Then an ellipse fitting technique is applied to detecting the circular sign. Further, a novel technique to validate the circular sign is proposed by employing the histogram of oriented gradient (HOG) which is calculated on the outer area of the sign. The experimental results show that the true positive rate of 0.92 and the precision of 0.97 could be achieved. The execution time of the proposed algorithm which is implemented on a Raspberry Pi Type 3 module is 48.17 ms. It provides a promising result for the real-time implementation.*

Keywords: Traffic sign detection, Embedded system, Color thresholding, Histogram of oriented gradient

1. **Introduction.** Traffic sign recognition (TSR) is an automatic system to recognize traffic signs, which is an important part in the driver assistance system (DAS). TSR is usually divided into detection and classification stages. The traffic sign detection is the most critical step and complicated task [1]. It localizes and detects the position of traffic signs in an image. Several methods have been proposed to detect traffic signs such as using color segmentation [2-4], shape information [5-9], color and shape information [10-20], color and Haar feature [21,22], histogram of oriented gradient (HOG) feature [23], and maximally stable extremal region (MSER) [24-27].

In the color segmentation method, traffic sign is detected based on the dominant color of the sign, i.e., red, blue, yellow, etc. In this method, the color space used in the segmentation process is the most important issue. Since the image captured by the camera is usually in RGB color space, it could be used directly without the color conversion process [2]. However, it is sensitive to the illumination changes. Therefore, the HSV color space which is insensitive to the illumination changes was employed in [3].

In the shape-based traffic sign detection, the shape of traffic sign is detected from the edge or gradient image which is extracted from the grayscale image. To detect the shape, several methods are employed such as voting the gradient [5], rectangle pattern matching and circle detection using pixel local direction voting [6]. The method in [5] is based on the fact that the direction of gradient of pixels in the circular or rectangular shapes will generate the highest vote on the center of shape. It first employed the Sobel operator to find the gradient vector of a pixel. Then the voting process was conducted by generating the voting lines of the pixels and accumulating the votes. The centers of signs were found at the pixels with the highest votes. In [6], a rectangle pattern matching was employed to find the traffic sign coarsely using a scan window. The scan window has four areas on the middle-top part, four areas on the middle-bottom part, four areas on the middle-left part, and four areas on the middle-right part. In each part, two areas on the outer sides are called as the black areas, while the ones on the inner sides are called as the white areas. The matching process was conducted by calculating the differences in brightness of the corresponding black and white areas. The method detected both rectangle and circular signs. To detect the circular sign, the pixel local direction voting was employed. It matched the pixel direction with the local border templates according to the position in the scan window, i.e., the directions of left, right, up, down, down-right, down-left, up-right, up-left.

The methods to combine color and shape information are commonly used. First, it employs the color segmentation methods to define the region of interest (ROI) of detected traffic signs. Then traffic signs are detected based on their shapes using several methods such as matching the geometry of shapes [10,13,14,18,19], circular Hough transform (CHT) [11,12], support vector machine (SVM) [15], and geometric fragmentation [17]. In [11], a segmented image was obtained by subtracting the gray scale image from the red component of RGB color image. It reduced the non-traffic sign objects and kept the objects that were similar to the traffic signs. Then the edge image was extracted using the Canny edge detector. Finally, the CHT was employed to find the circular sign in the edge image. In [15], the blob of candidate sign was obtained using the thresholding technique on the hue and saturation components of the HSI color space. Then the linear SVM was employed to classify the blob according to the shape. They proposed the feature vectors by calculating the distances from the external edge of the blob to the bounding box as the input to SVM. The feature vectors consist of four distance vectors in the left, right, upper, and bottom side. The feature was invariant to scale, rotation, and translation.

Other approaches to detect traffic signs are by employing the Haar feature [21,22] and the HOG feature [23]. In [21], the Haar feature trained with the Adaboost classifier was employed to detect traffic signs. In this approach, the color segmentation based-on HSV color space was firstly employed to define the ROI of traffic signs. Then the Adaboost classifier was employed to find the traffic signs in the ROI. In [22], instead of the HSV color space, the Haar feature was extracted from the modification of RGB color called as the RBAT (red-blue angle transformation). It exploited the red, blue and white colors of the traffic signs. Thus, it considered the red and blue channels (RB plane) to emphasize the traffic sign objects while omitting the non-traffic sign objects. The method first extracted the red and blue components of the RGB channel. It will project the three color channels of a pixel into two color channels in the RB plane. Then an angle of the pixel in the RB plane with respect to the blue axis was calculated. The angle was normalized into 8-bit value to represent a single color channel of RBAT. The RBAT images demonstrated the better results in distinguishing the red, blue and white traffic signs from the background compared to the gray scale images.

In [23], the HOG feature trained with SVM was employed to detect the warning signs. In the method, one SVM was trained to detect all types of warning signs. Thus, it could detect the warning signs in one scanning time. The MSER, a region where the shape remains the same when the image is thresholded over the different levels, was employed to detect traffic sign [24-27]. The connected components in MSERs were further filtered according to the width, height, etc., to reduce non-traffic sign objects. Since the stable region is obtained over the several threshold values, it is robust against the illumination changes. The effectiveness of MSER is affected by the color spaces used in the method. In [24], the grayscale image was employed to detect the traffic signs with the white background, while the normalized red/blue image was employed to detect the traffic signs with the red or blue color. The HSV color space was used in [26] to detect the red and blue traffic signs. In [27], they developed a color probability model from the sample traffic sign images and used the probability map in the MSER extraction. The major drawback of MSER is the high computation cost due to the iteration process in finding the stable regions. To reduce the execution time, the threshold values were limited in a specific range [24].

One important aspect that should be considered in the traffic sign detection is the simplicity and fast processing time, due to the requirement of real-time implementation in the DAS. Most of works discussed previously implemented the algorithms on the personal computer. Only a few works implemented the algorithms on the embedded platforms, such as using Xilinx FPGA [4,6], Freescale i.MX 6 [5], Raspberry Pi [8], and FriendlyARM Tiny4412 board [18].

In this paper, we propose an efficient traffic sign detection which is implemented on the low cost Raspberry Pi module. The proposed algorithm consists of color segmentation, shape detection, and sign validation. A novel method to validate the circular traffic sign is introduced. It uses the HOG feature on the outer side of detected sign for validating the sign. Using this validation step, the false positive detection could be reduced significantly. Compared to the existing methods [15,21-23], which require many sample images for training the classifiers (SVM, Adaboost), our method only need one sample image for sign validation. In the matching process, our method offers the faster execution time compared to the existing method proposed by [6]. The matching process employed in our sign validation method is performed on the sign candidates only, while the matching process in [6] is performed on a whole image by moving the scan window. Compared to the MSER method [24-27], where the image thresholding is processed in several iterations, our method only processes it in one iteration, which yields the faster computation time.

Since our method is implemented on a single board computer system, it could be developed easier compared to the FPGA system [4,7], due to the wide availability of the low cost single board computer systems.

The main contribution of our work is threefold. Firstly, it employs the efficient color thresholding to provide the sign candidates, which are further detected by an ellipse detection method. Secondly, it validates the detected sign using a novel and simple method. Thirdly, it implements the algorithm on a low cost embedded system.

The rest of paper is organized as follows. Section 2 presents our proposed system. Section 3 discusses the experimental results. Conclusion is covered in Section 4.

2. **Proposed System.**

2.1. **System overview.**

In this paper, we deal with the red circular traffic signs, where the outer border is painted with red color ring. Although the algorithm works with the red circular traffic sign, it could be extended to cope with the other types of traffic signs. The algorithm is divided into detection stage and validation stage as illustrated in Figure 1.

In the detection stage, the location of sign in an image is found by applying the color thresholding, the contour finding, and the ellipse fitting. This stage yields the bounding boxes of sign candidates. It is a common problem in the detection stage that the non-traffic sign objects are also found, called as the false positive. To discard those objects, we propose to validate the sign candidates in the validation stage.

In the validation stage, the HOG feature is employed as descriptor to match the sign candidates with the reference sign. Since there are many types of red circular signs, each candidate should be matched with all red circular reference signs. However, it will

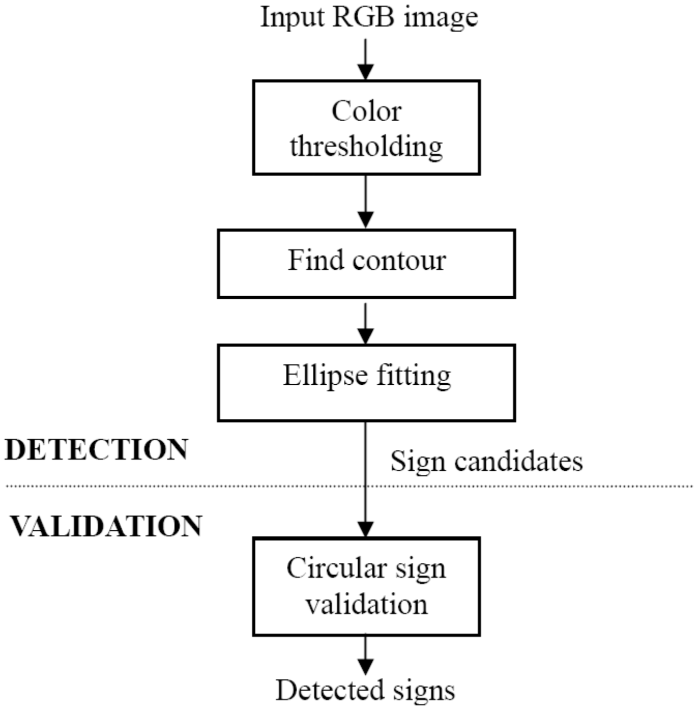


FIGURE 1. Flowchart of proposed method

consume time. Fortunately, the red circular signs appear with the same red color ring on the outer border. Therefore, to reduce the computation time, only one reference sign is used in the matching by taking account of the outer ring only. The detail of method is described in the next section.

2.2. **Sign detection.** The first step in the detection stage is the color thresholding. Since the red circular sign is considered, the red color thresholding is employed. In this work, we adopt and compare two red color thresholding techniques proposed in [22] and our previous method in [17].

In [22], a new color transformation called RBAT is defined as follows. Let $R(x, y)$ and $B(x, y)$ be the values of red and blue components of a pixel (x, y) in RGB color space, respectively. The value of RBAT color transformation is defined as $AT(x, y)$ and expressed by

$$AT(x, y) = \frac{255}{90} \times \tan^{-1}\left(\frac{R(x, y)}{B(x, y)}\right) \tag{1}$$

A pixel is assigned as red when $AT(x, y) > \mathit{Thr\_RBAT}$, where $\mathit{Thr\_RBAT}$ is a threshold which is defined empirically.

In [17], a normalized red component of a pixel (x, y) is defined as $r(x, y)$ and expressed by

$$r(x, y) = 255 \times \left(\frac{R(x, y)}{R(x, y) + G(x, y) + B(x, y)}\right) \tag{2}$$

where $R(x, y)$, $G(x, y)$ and $B(x, y)$ are the values of red, green and blue components of a pixel (x, y) in RGB color space, respectively. A pixel is assigned as red when $r(x, y) > \mathit{Thr\_norm\_red}$, where $\mathit{Thr\_norm\_red}$ is a threshold which is defined empirically.

In the second step, the contours of binary image obtained from the color thresholding are found using the border following algorithm proposed in [28]. The contours represent the borders of traffic signs. To verify that the object is a circular traffic sign, an ellipse fitting method [29] is applied to the pixels of the detected contour. Once the ellipse is detected, it is judged as the circular sign when the ratio of major and minor axes is below a certain value (i.e., 1.3 in this work).

2.3. **Sign validation.** In some conditions, traffic signs which are detected from the previous detection stage may contain non-traffic sign objects, due to the bad color thresholding and/or shape detection as illustrated in Figure 2. In the figure, two objects are detected as the traffic signs. The detected sign in the center of image is a true positive, where the speed limit sign is detected properly, while the detected sign in the right part of image is the false positive, since it detects the tree, which is not the traffic sign. It means that the sign detection described previously is unable to verify the circular traffic sign properly. To overcome this drawback, we add the validation stage using a simple effective method to keep the low computation time.

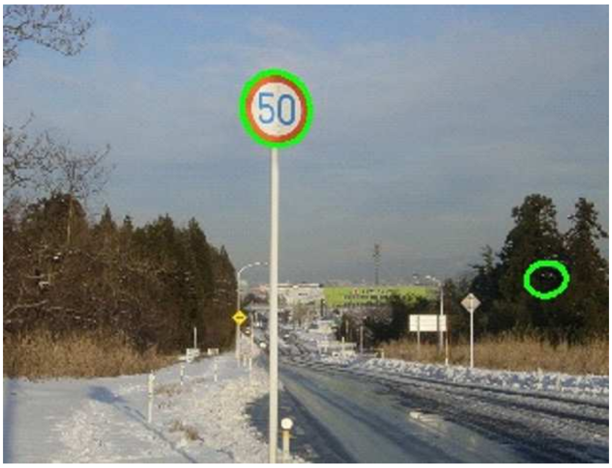


Figure 2. False traffic sign detection

In the traffic sign detection method using the color segmentation followed by the shape detection which is adopted in our proposed system, the selection of threshold (*Thr_RBAT* or *Thr_norm_red*) in the color segmentation is an important task. The threshold value affects the number of binary objects (blobs) that are fed to the ellipse fitting stage. When many blobs are generated, the ellipse fitting method may extract both the traffic sign and the non-traffic sign objects. However, when a few blobs are generated, the traffic sign objects may not be extracted. In our approach, we set the threshold value in such that many blobs are generated. Then to discard the non-traffic sign objects which are extracted by the ellipse fitting method, we introduce the validation step. This approach increases the true positive detection, while reduces the false positive detection. However, by introducing the sign validation step, the execution time will increase. Therefore, the sign validation technique should be fast and effective.

A common method to verify the circular traffic sign is by calculating the aspect ratio and the size of sign candidate. However, due to the simple rules employed in this validation process, the method often fails to verify the sign. Further, the size restriction requires a proper selection of the lower limit and upper limit sizes of the signs. The other method using template matching may be adopted. However, it requires many templates and it is usually considered as the sign classification stage. In this work, we introduce a novel technique to validate the circular sign by utilizing the outer ring of the sign as described in the following.

The basic idea behind our validation stage is that the red circular signs have the common red-color ring on the outer border. Thus, the aim of validation stage is to find this red ring. A simple way is to perform the template matching technique. To provide a robust matching under varying illumination, the HOG feature is employed in the matching. It is noted here that the matching is performed on the outer ring area only, not a whole image.

The HOG feature is first introduced in [30] to detect the pedestrian. It describes the histogram of orientation of gradient in an image as illustrated in Figure 3. In the validation stage, the detected sign is resized into 40 × 40 pixels. To calculate the HOG feature, the image is divided into 8 × 8 cells, where the cell's size is 5 × 5 pixels. The orientation histogram is calculated based on the overlapped blocks, where each block consists of 2×2 cells. In a block, the magnitude and orientation of the gradient of a pixel are calculated as follows. Let $G_h(x, y)$ and $G_v(x, y)$ be the gradients in horizontal and vertical directions of a pixel at (x, y) respectively. The image intensity of a pixel at (x, y) is expressed as $I(x, y)$. The gradients are computed as

$$G_h(x, y) = I(x + 1, y) - I(x - 1, y) \tag{3}$$

$$G_v(x, y) = I(x, y + 1) - I(x, y - 1) \tag{4}$$

The magnitude of gradient of a pixel at (x, y) is defined as $G(x, y)$ and expressed by

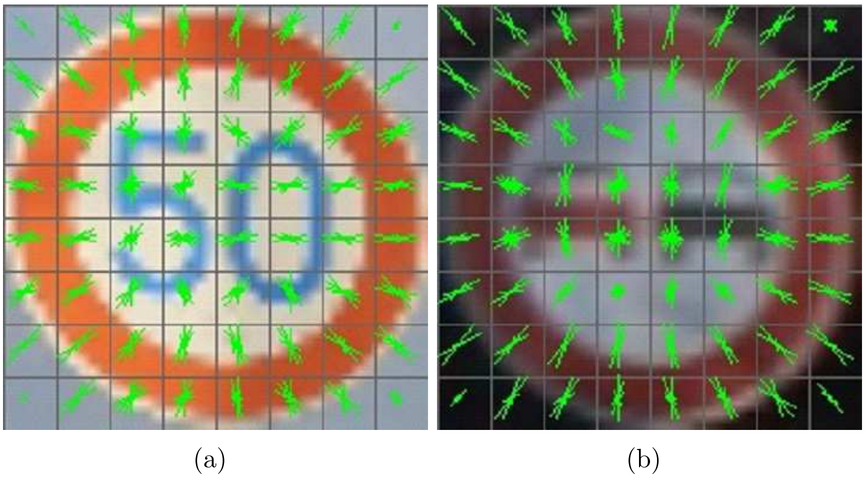
$$G(x, y) = \sqrt{G_h(x, y)^2 + G_v(x, y)^2} \tag{5}$$

The orientation of gradient of a pixel at (x, y) is defined as $\theta(x, y)$ and expressed by

$$\theta(x, y) = \tan^{-1} \frac{G_v(x, y)}{G_h(x, y)} \tag{6}$$

In order to create the histogram, the orientation is quantized into 9 bins and the vote which is assigned to the bin, is calculated based on the magnitude of gradient.

Figures 3(a) and 3(b) show two red circular signs and the visualization of their HOG features. They are different in the sign information and the brightness of image, where the color of the speed limit sign (Figure 3(a)) is brighter than the no overtaking sign in Figure 3(b). However, by examining the figures, we could see that the HOG features surrounding the red outer ring are almost similar, i.e., they represent the orientation of gradient of the circle or ring. It is noted that since image gradient reflects the directional change of the color in the image, the orientation of gradient shown in the figure is perpendicular to the circumference of circle.



(a) (b)

Figure 3. HOG feature of traffic sign image: (a) speed limit sign; (b) no overtaking sign

In the validation process, the sign candidate is validated as the red circular sign when it matches with the reference sign. Let us define the *outer_area* as the cells where the HOG feature will be calculated. It is denoted by the cell with dark color as shown in Figure 4. The procedure to match the sign candidate is as follows.

- Compute HOG feature of a reference image sign (could be any arbitrary red circular signs) in the *outer_area*. Save them as the HOG reference (called as *HOG_ref*).
- Compute HOG feature of the sign candidate in the *outer_area* (called as *HOG_candidate*).
- Compute the Euclidean distance between *HOG_ref* and *HOG_candidate*.
- If the distance is less than a threshold (*Thr_match*), then assign the candidate sign as the detected sign.

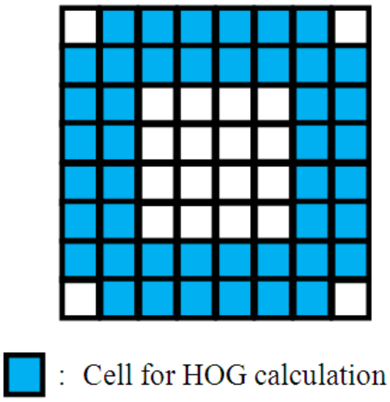


Figure 4. Area of cells for HOG calculation

3. **Experimental Results.** The proposed algorithm is implemented on a Raspberry Pi Type 3, the low cost single board computer equipped with 1.2 GHz 64-bit quad-core ARMv8 CPU, 1GB RAM. The C++ programming language and OpenCV library [31] are used to implement the algorithm, which is run on the Raspbian operating system [32]. To verify our method, several experiments are conducted. In the experiments, two color thresholding methods, i.e., RBAT and normalized red component, are evaluated. Further, the values of color thresholding threshold (*Thr_RBAT*, *Thr_norm_red*) and matching threshold (*Thr_match*) are varied to observe the effects of those parameters. They are evaluated in both approaches, i.e., without sign validation process and with sign validation process (our proposed method).

We also compare our method with the existing technique using the MSER proposed in [24]. In the original work [24], the stable regions obtained by the MSER are further sorted according to the width, height, aspect ratio, region perimeter and area, and bounding-box perimeter. Since those parameters depend on the sizes of traffic signs in the images, which differ with our image datasets, we could not adopt the parameters directly. Instead, the stable regions are further sorted using the same ellipse fitting and the shape criteria that are employed in our proposed method.

The image datasets used in the experiments consist of 100 images of the German red circular traffic signs, which are taken from the German Traffic Sign Detection Benchmark [16] (called as **GDataset**) and 100 images of the Japanese red circular traffic signs (called as **JDataset**). The image size in **GDataset** is 1360×800 pixels, where the sizes of traffic signs vary from 48×48 to 128×128 pixels. The image size in **JDataset** is 1280×960 pixels, where the sizes of traffic signs vary from 80×80 to 300×300 pixels. Sample images

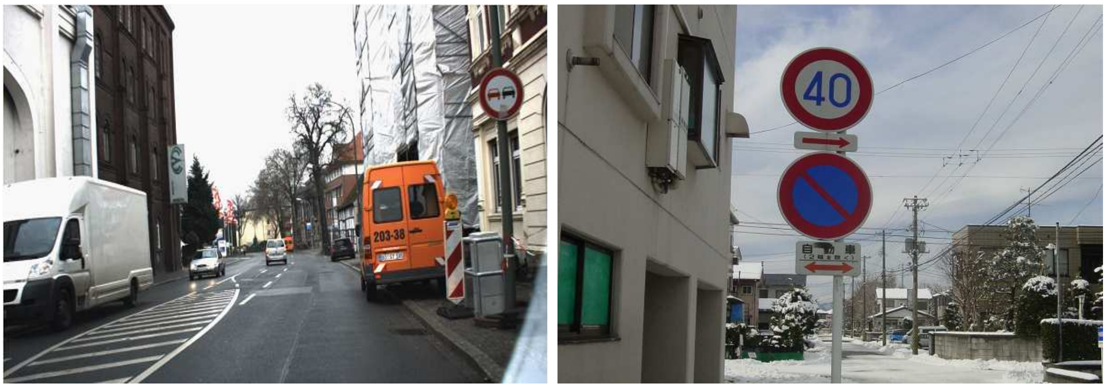


Figure 5. Sample images in **GDataset** (left) and **JDataset** (right)

from two datasets are shown in Figure 5, where the image in left side is from **GDataset**, while the image in right side is from **JDataset**. The number of red circular signs in each dataset varies from one to three signs. During experiments, the image's height is resized into 240 pixels and the image's width is resized proportionally.

To evaluate the effectiveness of our proposed detection, **TPR** (true positive rate) and **PREC** (precision) are calculated. **TPR** is defined as ratio of the number of true positives to the total number of signs in the image datasets. **PREC** is defined as ratio of the number of true positives to the total number of true positives and false positives.

3.1. **Results of proposed method.**

The detection results using the RBAT color thresholding are given in Table 1 and Table 2. In Table 1, the value of *Thr_RBAT* is varied, while the value of *Thr_match* is set to 0.35. It is clearly shown from the table that **TPR** and **PREC** in both methods (without and with sign validation) are affected by the value of *Thr_RBAT*. The highest **TPR**s in both methods are achieved when the value of *Thr_RBAT* is 150. It is also shown that **TPR** of the method without sign validation is higher than the method with sign validation. This result could be understood from the fact that in the method without sign validation, all detected sign candidates are considered as the traffic signs. Thus, it yields the high true positives and the high false positives. Therefore, **TPR** is high and **PREC** is very low as shown in the table. On the contrary, when sign validation is applied, the wrong detected objects will be discarded; thus the false positives are decreased significantly. Unfortunately, the sign validation decreases **TPR** slightly due to the performance of matching technique employed in the sign validation as discussed in the following.

Table 1. Detection results using RBAT color thresholding when *Thr_RBAT* is varied, while *Thr_match* is set to 0.35

RBAT color thresholding (*Thr_match* = 0.35)				
Thr_RBAT	**Without validation**		**With validation**	
	TPR	**PREC**	**TPR**	**PREC**
130	0.76	0.27	0.55	0.96
140	0.85	0.33	0.76	0.99
150	**0.98**	**0.37**	**0.91**	**0.99**
160	0.88	0.42	0.83	1.00
170	0.73	0.50	0.69	0.99

Table 2. Detection results using RBAT color thresholding when *Thr_match* is varied, while *Thr_RBAT* is set to 150

RBAT color thresholding ($\boldsymbol{Thr\_RBAT} = \mathbf{150}$) with validation		
Thr_match	**TPR**	**PREC**
3.50	0.91	0.99
3.75	**0.92**	**0.96**
4.00	0.93	0.89

Table 3. Detection results using normalized red color thresholding when *Thr_norm_red* is varied, while *Thr_match* is set to 0.35

Normalized red color thresholding ($\boldsymbol{Thr\_match} = \mathbf{0.35}$)				
Thr_norm_red	**Without validation**		**With validation**	
	TPR	**PREC**	**TPR**	**PREC**
90	0.82	0.29	0.68	1.00
100	**0.92**	**0.56**	**0.89**	**0.99**
115	0.88	0.68	0.82	0.99

Table 4. Detection results using normalized red color thresholding when *Thr_match* is varied, while *Thr_norm_red* is set to 100

Normalized red color thresholding ($\boldsymbol{Thr\_norm\_red} = \mathbf{100}$) with validation		
Thr_match	**TPR**	**PREC**
3.50	0.89	0.99
3.75	**0.92**	**0.97**
4.00	0.92	0.93

To examine the effectiveness of sign validation, we vary the value of *Thr_match*, while the value of *Thr_RBAT* is set to 150, i.e., the optimal value of *Thr_RBAT* obtained from Table 1. The detection results are given in Table 2. From the table, it is obtained that by increasing the value of *Thr_match*, **TPR** increases, but it also decreases **PREC**. It means that when the threshold (*Thr_match*) for matching the candidate sign is increased, the number of candidates that is validated as true traffic sign (true positive) is increased. At the same time, it also increases the false positive that leads to decreasing **PREC**. From Table 2, the optimal value of *Thr_match* is 0.375, which yields **TPR** of 0.92 and **PREC** of 0.96.

The detection results using the normalized red color thresholding are given in Table 3 and Table 4. Similar to previous tables, Table 3 gives the results when the value of *Thr_norm_red* is varied, while the value of *Thr_match* is set to 0.35. Table 4 gives the results when the value of *Thr_match* is varied, while the value of *Thr_norm_red* is set to 100, which is the optimal value of *Thr_norm_red* obtained from Table 3.

Examining Table 3 and Table 4, the effects of *Thr_norm_red* and *Thr_match* exhibit the similar conditions to the RBAT color thresholding discussed previously. It is obtained that the optimal values of **TPR** and **PREC** are 0.92 and 0.97 respectively. The results are almost the same as the ones achieved by the RBAT color thresholding.

To see the effectiveness of our proposed method and the effects of the parameters easily, we plot **TPR** versus **PREC** curve as shown in Figure 6. In the figure, *RBAT_wo_vt_0.35*

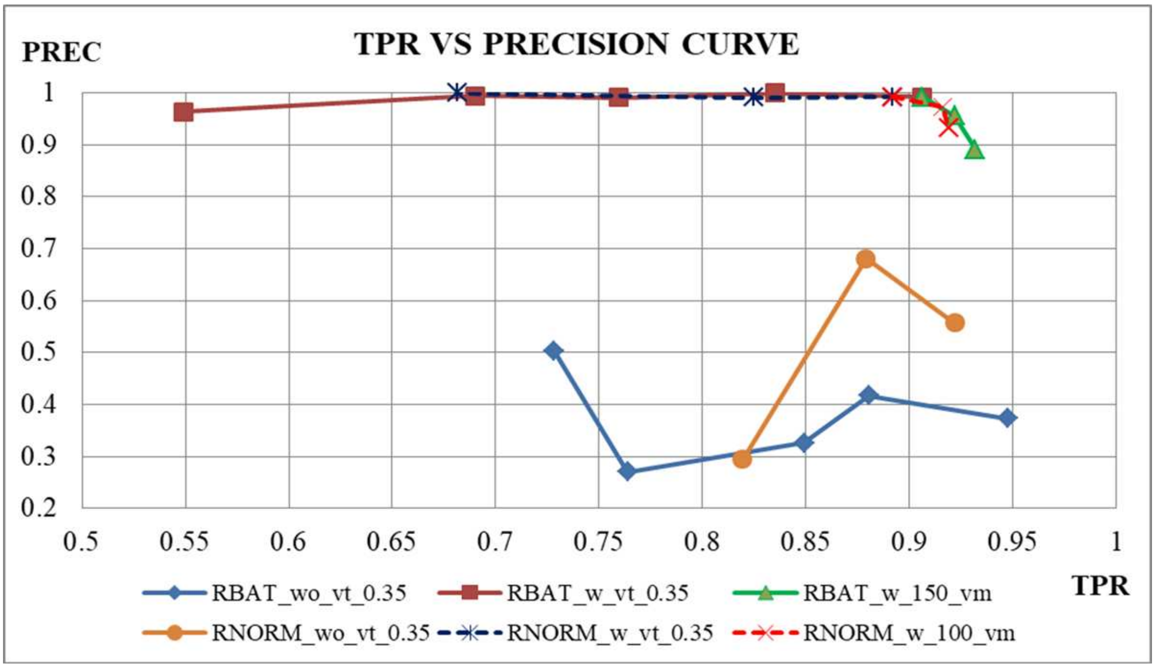


Figure 6. **TPR** versus **PREC** curve

and *RBAT_w_vt_0.35* curves denote the RBAT color thresholding when *Thr_RBAT* is varied while *THR_match* is set to 0.35, without and with sign validation respectively. *RBAT_w_150_vm* curve denotes the RBAT color thresholding when *Thr_RBAT* is set to 150 while *THR_match* is varied, with sign validation. *RNORM_wo_vt_0.35* and *RNORM_w_vt_0.35* curves denote the normalized red color thresholding when *Thr_norm_red* is varied while *THR_match* is set to 0.35, without and with sign validation respectively. *RNORM_w_100_vm* curve denotes the normalized red color thresholding when *Thr_norm_red* is set to 100 while *THR_match* is varied, with sign validation.

From the figure, the curves of method with sign validation lie on the top of the ones without sign validation. It proves that the sign validation process increases the precision (**PREC**) effectively. Let us examine the upper curves that represent the results of method with sign validation. Both the RBAT and normalized red color thresholding techniques exhibit the similar results, i.e., when the color thresholds (*Thr_RBAT* and *Thr_norm_red*) are varied and the matching threshold (*Thr_match*) is kept constant, **TPR**s increase until the maximum values are achieved (0.91 for the RBAT color threshoding and 0.89 for the red color thresholding), while **PREC**s change slightly. Further, from these maximum points, when the color thresholds are kept constants and the matching threshold is varied, **TPR**s increase but **PREC**s decrease so that the lines form the bends as shown in the top right of the figure. Therefore, the optimal results (high **TPR**s and high **PREC**s) could be found at the corners of the bends.

From above discussion, it is worthy noting that besides producing high **TPR** and high **PREC**, our proposed method also provides a systematic way to find the optimal result of **TPR** and **PREC** as follows. At first, while the matching threshold is constant, we vary the color threshold and find the value that produces the highest **TPR**. Then using this color threshold, we vary the matching threshold and find the value that produces the optimal value of **TPR** and **PREC** by examining the corner of bend.

To examine the execution time of our proposed algorithm which is implemented on an embedded system, we measure the execution time of the RBAT color thresholding and the normalized red color thresholding, with sign validation. The measurement of execution

TABLE 5. Execution time of proposed method

Execution time (ms)				
Method	**Color conversion**	**Shape finding**	**Validation**	**Total**
RBAT (*Thr_RBAT* = 150; *Thr_match* = 3.75; with validation)	45.24	9.55	6.46	61.25
Normalized Red (*Thr_norm_red* = 100; *Thr_match* = 3.75; with validation)	37.81	6.05	4.31	48.17

time is divided into color conversion, shape finding, and sign validation. The results are given in Table 5. From the table, it is obtained that the total execution time using the RBAT color thresholding is 61.25 ms (or 16.33 fps if it is employed to process the video), while the total execution time using the normalized red color thresholding is 48.17 ms (or 20.76 fps if it is employed to process the video). The results suggest that the proposed method is suitable for the real-time implementation.

Some image examples of detection results from **GDataset** and **JDataset** are shown in Figure 7. In the figure, the ellipse denotes the detected traffic sign without sign validation, while the rectangle denotes the detected traffic sign with sign validation. The images in the first row show the condition where both methods (with and without sign validation) generate true positive without false positive. The images in the second row show the condition where the method with sign validation generates true positive without false positive, while the method without sign validation generates true positive and false positive. The images in the third row show the condition where both methods, with and without validation, generate true positive and false positive. Many false positives occurring in the background images shown in the second rows clarify the result of low **PREC**s which are produced by the methods without sign validation as given in Table 1 and Table 3.

3.2. Results of existing method (MSER).

The experiments of existing method using MSER are conducted with four different values of *Delta*, i.e., the step size of threshold value to be changed during the process for finding the stable region. Increasing the value will result in less regions and decrease the computation time. In the experiment, the normalized red color space is employed as the image input of MSER method. The experimental results are listed in Table 6, where the values of *Delta* are 2, 5, 10, and 20.

From the table, it is obtained that when *Delta* increases, **TPR** decreases while **PREC** increases. The results are similar to the methods without sign validation as discussed previously. The optimal result is obtained when *Delta* of 10 is employed, which yields **TPR** of 0.88 and **PREC** of 0.77. The comparison results of our proposed method and the MSER method are given in Table 7, where the result of each method is the optimal one. It is obvious that our method is superior to the MSER method, i.e., **TPR** and **PREC** are higher and the execution time is faster. The superiority of our method lies on the sign validation method, which reduces the false positive effectively with the fast processing time. Referring to Table 5, the execution time of sign validation method is about 10% from the total execution time.

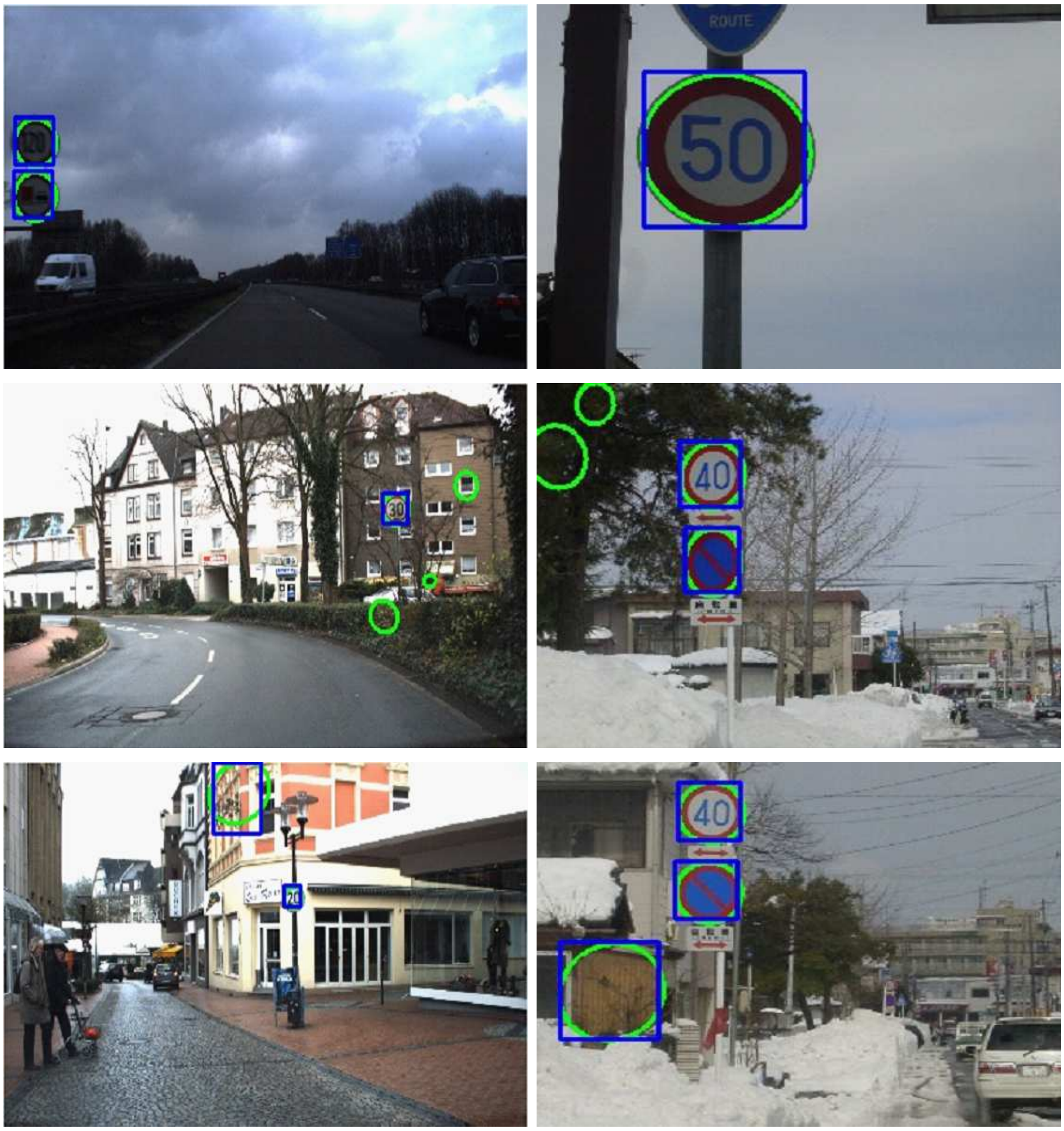


Figure 7. Examples of detection results from **GDataset** and **JDataset**

Table 6. Detection results using the MSER method

Delta parameter	TPR	PREC	Execution time (ms)		
			Color conversion	MSER	Total
Delta = 2	0.97	0.33	37.67	69.38	107.05
Delta = 5	0.97	0.58	37.77	52.95	90.72
Delta = 10	0.88	0.77	37.77	48.09	85.86
Delta = 20	0.59	0.86	37.79	45.80	83.59

4. **Conclusions.** An efficient algorithm was implemented on an embedded system for detecting red circular traffic signs. Both color and shape information was employed to detect the signs. Further, a sign validation process was introduced to validate the detected sign candidates. The proposed method was evaluated by varying the parameters used by the color thresholding step and the sign validation step. The proposed method achieved the high true positive rate and the high precision, while the computation time is low,

Table 7. Comparison results of our proposed method and the MSER method

Method	TPR	PREC	Execution time (ms)
Proposed method – RBAT	0.92	0.96	61.25
Proposed method – Normalized Red	0.92	0.97	48.17
MSER method	0.88	0.77	85.86

which satisfied the real-time implementation. Compared with the MSER method, our proposed method achieved the higher true positive rate and the precision, and the lower computation time. In future, the method will be extended to cope with the other types of traffic signs. Furthermore, the proposed system will be tested in the real environments.

Acknowledgment. This work is supported by the Research Grant 2017, Competence-Based Research scheme from Directorate General of Higher Education, Ministry of Research and Technology and Higher Education, Republic of Indonesia, No.: SP DIPA-042.06.1.401516/2017.

REFERENCES

[1] A. Møgelmose, M. M. Trivedi and B. Moeslund, Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey, *IEEE Trans. Intell. Transp. Syst.*, vol.13, no.4, pp.1484-1497, 2012.

[2] T. Surinwarangkoon, S. Nitsuwat and E. J. Moore, Traffic sign recognition by color filtering and particle swarm optimization, *Proc. of the 4th International Conference on Computer Research and Development*, Chengdu, China, pp.55-59, 2012.

[3] N. B. Romdhane, H. Mliki and M. Hammami, An improved traffic signs recognition and tracking method for driver assistance system, *Proc. of 2016 IEEE/ACIS the 15th Int. Conf. on Computer and Information Science*, Okayama, Japan, pp.1-6, 2016.

[4] S. Waite and E. Oruklu, FPGA-based traffic sign recognition for advanced driver assistance systems, *J. of Transp. Technol.*, vol.3, no.1, pp.1-6, 2013.

[5] Y. T. Lin, T. Chou, M. S. Vinay and J. I. Guo, Algorithm derivation and its embedded system realization of speed limit detection for multiple countries, *Proc. of 2016 IEEE International Symposium on Circuits and Systems*, Montreal, Canada, pp.2555-2558, 2016.

[6] A. T. Hoang, T. Koide and M. Yamamoto, Low cost hardware implementation for traffic sign detection system, *Proc. of 2014 IEEE Asia Pacific Conference on Circuits and Systems*, Ishigaki, Japan, pp.363-366, 2014.

[7] R. Laguna, R. Barrientos, L. F. Blazquez and L. J. Miguel, Traffic sign recognition application based on image processing techniques, *Proc. of the 19th World Congress of the International Federation of Automatic Control*, Cape Town, South Africa, pp.104-109, 2014.

[8] E. Bilgin and S. Robila, Road sign recognition system on Raspberry Pi, *Proc. of 2016 IEEE Long Island Systems, Applications and Technology Conference*, New York, USA, pp.1-5, 2016.

[9] M. A. García-Garrido, M. Ocana, D. F. Llorca, E. Arroyo, J. Pozuelo and M. Gavilan, Complete vision-based traffic sign recognition supported by an I2V communication system, *Sensors*, vol.12, no.2, pp.1148-1169, 2012.

[10] S. B. Wali, M. A. Hannan, A. Hussain and S. A. Samad, An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and SVM, *Math. Prob. in Eng.*, vol.2015, no.250461, pp.1-11, 2015.

[11] R. Biswas, H. Fleyeh and M. Mostakim, Detection and classification of speed limit traffic signs, *Proc. of 2014 World Congress on Computer Applications and Information Systems*, Hammamet, Tunisia, pp.1-6, 2014.

[12] A. Adam and C. Ioannidis, Automatic road-sign detection and classification based on support vector machines and HOG descriptors, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol.2, no.5, pp.1-7, 2014.

[13] T. Bui-Minh, O. Ghita, P. F. Whelan and T. Hoang, A robust algorithm for detection and classification of traffic signs in video data, *Proc. of 2012 International Conference on Control, Automation and Information Sciences*, Ho Chi Minh City, Vietnam, pp.108-113, 2012.

[14] J. Miura, T. Kanda, S. Nakatani and Y. Shirai, An active vision system for on-line traffic sign recognition, *IEICE Trans. Inf. and Syst.*, vol.E00-A, no.11, pp.1-8, 2002.

[15] S. M. Bascon, S. L. Arroyo, P. G. Jimenez, H. G. Moreno and F. L. Ferreras, Road-sign detection and recognition based on support vector machines, *IEEE Trans. Intell. Transp. Syst.*, vol.8, no.2, pp.264-278, 2007.

[16] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing and C. Igel, Detection of traffic signs in real-world images: The German traffic sign detection benchmark, *Proc. of the 2013 International Joint Conference on Neural Networks*, TX, USA, pp.1-8, 2013.

[17] A. Soetedjo and K. Yamada, An efficient algorithm for traffic sign detection, *J. of Adv. Comput. Intell. and Intell. Inform.*, vol.10, no.3, pp.409-418, 2006.

[18] T. Q. Vinh, Real-time traffic sign detection and recognition system based on FriendlyARM Tiny4412 board, *Proc. of 2015 International Conference on Communications, Management and Telecommunications*, DaNang, Vietnam, pp.142-146, 2015.

[19] H. S. G. Supreeth and C. M. Patil, An approach towards efficient detection and recognition of traffic signs in videos using neural networks, *Proc. of 2016 International Conference on Wireless Communications, Signal Processing and Networking*, Chennai, India, pp.456-459, 2016.

[20] J. Darko and L. Sven, Warning and prohibitory traffic sign detection based on edge orientation gradients, *Proc. of the Croatian Computer Vision Workshop*, Zagreb, Croatia, 2014.

[21] L. Chen, Q. Li, M. Li and Q. Mao, Traffic sign detection and recognition for intelligent vehicle, *Proc. of 2011 IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, pp.908-913, 2011.

[22] Y. Gu, M. P. Tehrani, T. Yendo, T. Fujii and M. Tanimoto, Traffic sign recognition with invariance to lighting in dual-focal active camera system, *IEICE Trans. Inf. and Syst.*, vol.E95-D, no.7, pp.1775-1790, 2012.

[23] F. Zaklouta and B. Stanciulescu, Warning traffic sign recognition using a HOG-based K-d tree, *Proc. of 2011 IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, pp.1019-1024, 2011.

[24] J. Greenhalgh and M. Mirmehdi, Real-time detection and recognition of road traffic signs, *IEEE Trans. Intell. Transp. Syst.*, vol.13, no.4, pp.1498-1506, 2012.

[25] S. K. Kundu and P. Mackens, Speed limit sign recognition using MSER and artificial neural networks, *Proc. of 2015 IEEE the 18th International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, pp.1849-1854, 2015.

[26] W. Farhat, H. Faiedh, C. Souani and K. Besbes, Real-time recognition of road traffic signs in video scenes, *Proc. of 2016 the 2nd International Conference on Advanced Technologies for Signal and Image Processing*, Monastir, Tunisia, pp.125-130, 2016.

[27] Y. Yang, H. Luo, H. Xu and F. Wu, Towards real-time traffic sign detection and classification, *IEEE Trans. Intell. Transp. Syst.*, vol.17, no.7, pp.2022-2031, 2016.

[28] S. Suzuki and K. Abe, Topological structural analysis of digitized binary images by border following, *Comput. Vis. Graph. Imag. Process.*, vol.30, no.1, pp.32-46, 1985.

[29] A. W. Fitzgibbon and R. B. Fisher, A buyer's guide to conic fitting, *Proc. of the 5th British Machine Vision Conference*, Birmingham, England, pp.513-522, 1995.

[30] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp.886-893, 2005.

[31] *http://opencv.org/.*

[32] *https://www.raspbian.org/.*