Design and Implementation of Portable Embedded System for Real-Time Traffic Sign Detection and Recognition

Truong Quang Vinh

***Abstract*—Nowadays, driver-assistance system is much considered by many automotive companies for manufacturing moderns' cars. In that system, traffic sign detection and recognition algorithm is a challenge problem that many researchers try to solve. This paper presents a design and implementation of the portable real-time traffic sign detection and recognition (TSDR) to help drivers notify the traffic signs in the streets. The main features of the TSDR system are real-time processing capability and high accuracy. To achieve these targets, a fusion method which is combination of advanced techniques including adaptive chromatic color segmentation, shape matching, and support vector machine (SVM) is proposed. Besides, a multi-threading programming technique is applied to enhance the real-time processing capability of the system. The TSDR system is implemented on a portable embedded system board with ARM Cortex-A9 processor. The TSDR system has been tested on the streets of Ho Chi Minh city. The experiments show that the proposed system can detect and recognize the traffic signs with accuracy of 93% at 15 frames per second.**

***Index Terms*— Traffic sign, chromatic color segmentation, shape matching, support vector machine, multi-thread programming**

1 INTRODUCTION

TRAFFIC accident is a serious problem in most of contries, especially in Vietnam. Most of accidents are caused by faults of drivers. Accidents often occur when the drivers are distracted, tired, or stressed. Therefore, traffic sign detection and recognition (TSDR) system is very necessary to help drivers notice the road signs.

The algorithm for TSDR is considered by many researchers to improve the accuracy and performance. Typically, the algorithm is composed of three stages: preprocessing, detecting, and recognizing.

In the preprocessing stage, the image frame is processed to prepare input data for the detecting stage. The preprocessing methods can be included image enhancement, denosing, resizing, and color converting. This stage can reduce some potential issues which may occur in the detecting stage. For instance, S. Maldonado Bascon et al [1] proposed a preprocessing method including gray level normalization, contrast stretching, and equalization to enhance the input image quality and thus overcome issues of the variation of illumination conditions.

In the detecting stage, the image regions of traffic signs are detected and cropped. The common approach for this stage is based on color and shape of the traffic signs. For using the color feature, color models such as RGB, CIELab, and HSV have been attempted to apply for detecting traffic signs. Some authors applied HSV color model to overcome the issues of illumination, poor lighting, or bad weather condition. C.Y. Yang [2] et al used Hue component for the sensory component to analyze the degree of similarity between the candidate image region and the road sign. H. Fleyeh [3] proposed a color segmentation method for traffic signs, namely the Shadow and Highlight Invariant Color Segmentation, which takes advantage of the

Received: May 6th, 2018; Accepted: Sep 17th, 2018; Published: Dec 30th, 2018

Truong Quang Vinh was with Department of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology - Viet Nam National University - Ho Chi Minh City (e-mail: tqvinh@hcmut.edu.vn).

invariant properties of the hue component. By using difference color space, Luis David Lopez [4] et al proposed to use CIELab color space and Gaussian model to detect the traffic signs. For using the shape feature, some geometric methods have been proposed to determine triangular and circular shapes. P. Rosin presented a method for measuring shapes: ellipticity, rectangularity, and triangularity, namely Affine Moment Invariant [5]. This method has been used for detecting the triangular and circular traffic signs in the work of Thanh Bui-Minh et al [6]. Claw Bahlmann et al [7] proposed an approach to combine both color feature and shape of the road signs by using color sensitive Haar Wavelet feature obtained from Ada boost training.

In the recognizing stage, many researchers employ learning machine such as Neural Network (NN), Support Vector Machine (SVM), and Deep Learning to recognize the traffic signs. Fistrek T. et al [8] presented a TSDR method using NN ad histogram based selection of segmentation. Jack Greenhalgh et al [9] used SVM with histogram of oriented gradient (HOG) features. Thanh Bui-Minh et al [6] used binary pictograms for SVM to classify traffic sign candidate regions. Recently, deep learning method is much considered by researchers thanks can provide good performance for traffic sign recognition. Rong-Qiang Qian [10] et al applies convonlution neural network for traffic sign recognition and achieve 96% accuracy. However, deep learning method requires very high computation and thus it is difficult to adapt real-time processing capability.

In order to implement the whole system of TSDR, researchers have combined several approaches for each processing stages to achieve the best performance in term of accuracy and speed. Most of authors implemented the algorithm on PC which cannot be mounted directly on cars. Some proposed TSDR systems have been successfully built for real-time capability on PC [9], [11]. However, lack of works performs TSDR on a compact and portable system.

In this paper, a design and implementation of the real-time TSDR system based Friendly ARM Tiny4412 board is presented. The contribution of this paper is an effective algorithm for the TSDR system with high accuracy and real-time processing capability. In order to achieve the research target, a fusion method which is combination of advanced techniques including adaptive chromatic color segmentation, shape matching, and (SVM) is proposed. Besides, the real-time processing capability of the system is improved by applying the multi-thread programming technique. The final prototype design is tested on the real streets for practical applications.

2 SYSTEM DESIGN

The design of TSDR system is carried out according to the embedded system design process [9]. This process consists of 5 steps: system specification, system partitioning, hardware / software design, integration, and testing.

In the system specification, five documents including product specification, engineering specification, hardware specification, software specification, and test specification were developed. The details of these documents are described as follows.

2.1 *Product Specification*

The final product of this research is the implementation of the TSDR system. This system can capture the videos on the street, detect the traffic signs in the input image frames, recognize the types of traffic signs, and notify the drivers by displaying the result on the LCD and playing the voice. This system is required to be compact, portable, real-time processing, and highly accurate.

2.2 *Engineering Specification*

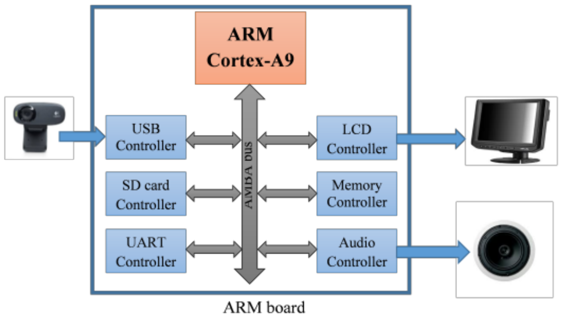


Fig 1. Block diagram of the TSDR system

The system consists of four parts: a camera, LCD, speaker, and an ARM board as shown in Fig. 1. The camera collects images and sends data to the embedded processor via the USB interface. Embedded system board processes image data from the camera, applies the detection and recognition algorithm, displays results on the LCD, and play notifying voice.

The system must satisfy some constrains described in Table 1. In order to decide the constraint values, the author considers the practical applications when the TSDR system is applied for the cars travelling in the city.

Table 1. Constraints of the TSDR system

Constraints	Value
Real-time processing	>15fps (frames per second)
Accuracy	> 90%
Number of traffic signs which can be recognized	50
Distance between traffic signs and the TSDR system which the system is able to detect traffic signs	>10m

The value 15fps for the real-time processing constraint means that the maximum delay to process 1 frame is 0.067s. The system needs at least 3 consequent frames to provide the reliable results, and thus the required delay is 0.2s. The typical maximum speed of cars in the city is 60 km/h, i.e. 16.67m/s. The images of traffic signs can be captured at the distance of 20m, and the system has to detect and recognize these signs at the distance of 10m in front the car. Therefore, the time period of the appearance of traffic signs is 0.6s when the car has moved 10m so far. The value 0.6s, which is a minimum delay time for the system to produce the results, is sufficient for the processing time 15fps. The system has 0.4s for the time budget in case the traffic signs are occluded or the cars move faster.

The accuracy constraint is chosen based on the recent research results for the TSDR algorithms which achieve at least 90%. This accuracy is enough to support the drivers to notify the traffic signs on the streets.

In this research, the author suggests the number of traffic signs which can be recognized is 50. These traffic signs which are popular in the city and thus the prototype system can be adapted for the real tests on the streets.

The minimum distance between traffic signs and the TSDR system which the system is able to detect traffic signs is 10m because the drivers need time to verify the appearance of the traffic signs, if necessary, and respond with the situations on the streets.

2.3 Hardware Specification

The hardware of TSDR system is based on the Friendly ARM Tiny4412 embedded board. This board is equipped with a quad-core ARM Cortex-A9 Exynos4412 processor 1.5GHz and 1GB SDRAM. The ARM processor supports USB controller to interface with the camera, LCD controller to display results on 7” TFT LCD, and audio controller to notify the driver by voice. An optical zoom is utilized to magnify the captured images from the camera. Therefore, the system can detect traffic signs from the distance of 10 meters.

2.4 Software Specification

The embedded software of the TSDR system includes the following functions:

•Reading image data from the camera.

•Displaying the images on the LCD and communicating with the users through the touch screen.

•Detecting and recognizing traffic signs by processing the captured images.

•Exporting audio signal to notify the users about the traffic signs.

In order to perform all these functions, a software structure including operating system (OS), libraries, and an application program is built.

The embedded Linux kernel is used for the OS. This kernel supports to manage the hardware resource and the application program. Especially, the Linux kernel supports multi-thread programming which can be applied for a multi-core processor. By using multi-thread, the power of a quad-core ARM processor can be utilized to perform the processing program in real-time.

The libraries for the software system are Qt Everywhere, Tslib, and OpenCV. Qt Everywhere

4.7 is used for designing the graphic user interface. Tslib is a library for controlling the touch screen. OpenCV is an open source library for computer vision. This library supports over 500 functions for many fields in computer vision such as image processing, pattern recognition, robot controlling etc. In this design, OpenCV 2.4 is used to implement image processing functions and SVM classifier of the TSDR algorithm.

2.5 *Test Specification*

The system will be tested in three procedures: hardware testing, software testing, and system testing.

The hardware test includes checking USB camera, checking LCD and touch screen, checking processor, checking peripherals.

The software test includes checking the Linux OS, checking the device drivers, and testing the software functions.

System test includes testing the system on the streets and evaluating the accuracy and real-time capability of the system.

3 PROPOSED TSDR ALGORITHM

The proposed TSDR algorithm includes three stages: pre-processing, detecting, and recognizing. The algorithm is optimized for the accuracy and processing time. Therefore, the low complexity techniques with high accuracy are preferred.

3.1 *Pre-processing*

In this stage, some pre-processing tasks including region selecting, color space conversion, white balancing, and de-noising are performed. Region selecting is to crop the region of interest (ROI) that is supposed to appear the traffic sign boards. In our implementation, the top-right region for ROI is chosen. The white balancing is to adjust the intensity of color of the input image for better view. This step makes the color segmentation in the detecting stage perform more robust.

3.2 *Detecting*

The objective of the detecting stage is to extract the candidate regions of traffic signs in the input image. The traffic signs are detected based on their color and shape characteristics as shown in the Table 2.

Table 2. Characteristics of the traffic signs in Vietnam

Types	**Characteristics**	**Samples**
Prohibitory signs	Red border, white or blue background, round shape	
Warning signs	Red border, black symbol on yellow background, triangle shape	
Guide signs	Blue background, white symbol, round shape	

According to the characteristics of each type of traffic signs, three processing steps consisting of chromatic color segmentation, refinement, and shape matching are applied to detect the candidate regions of traffic signs in the input image.

3.2.1 Chromatic color segmentation

The proposed color segmentation uses HSV (Hue-Saturation-Value) color model. The color in the outdoor images on the roads is very sensitive to variation of lighting condition. The conventional methods normally use Hue to segment the color regions. However, Hue becomes meaningless when Value is very low or very high. Besides, the color segmentation result is also unstable when the Saturation is very low. Therefore, Saturation and Value are used to determine the chromatic zone for the red as shown in Fig. 2. This chromatic zone is used for segmentation of the red region. A pixel which has the value belonging to the chromatic zone is recognized as the red region and be set to 1. The other pixels are set to 0.

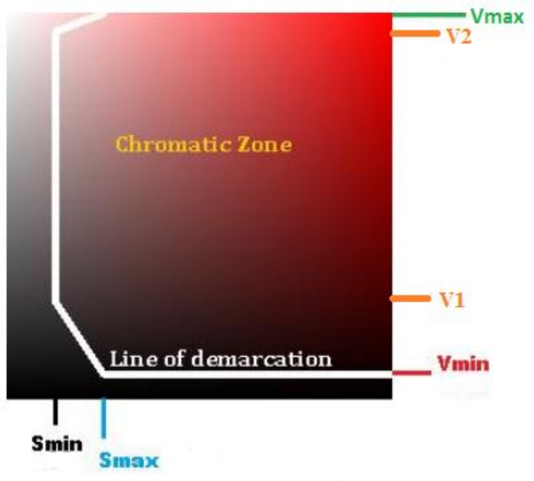


Fig 2. The chromatic zone for red color

By the similar way, the segmentation process is applied for the guide signs by using the chromatic zone for the blue color. The parameters for the chromatic zone are shown in Table 3. These parameters are chosen by experiments.

Table 3. Parameters for the chromatic zone

	Red	Blue
Hue (H)	[0:10] or [170:180]	[100:120]
Satuation (S)	Smin = 64; Smax = 96	

Value (V)	V1= 96; V2= 243
	Vmin= 64; Vmax= 255	

After applying color segmentation, we obtain a binary image in which the black pixels represent for the background and the white pixels represents for the candidate region.

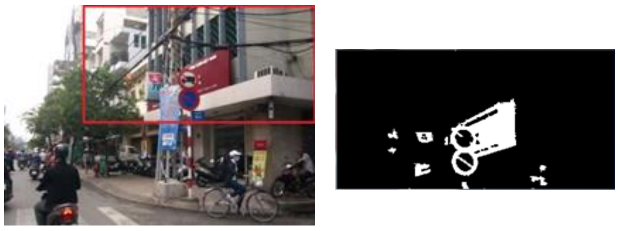


Fig. 3. The demonstration result of chromatic color segmentation.

The result image of color segmentation may contains the traffic signs and noise objects such as advertisement boards, flags, panel, etc. as shown in Fig. 3. In the next step, the refinement step is applied to separate the candidate regions from the noise objects in the background.

3.2.2 Refinement

In this step, the area of each candidate region is first calculated, and then the maximum and minimum thresholds are applied to eliminate the candidate regions which are either too large or too small. Next, the inner parts of the objects are taken by using the following steps:

i. Fill the background by white pixels
ii. Invert the pixel 0 and 1
iii. Apply dilation to fill disconnected lines inside the object regions.
iv. Apply maximum and minimum thresholds for the area of each object to eliminate the candidate regions which are either too large or too small.

Fig. 4 shows the demonstration of the refinement steps.

By taking the inner part of the objects, two traffic signs which are close together can be separated, and the noise objects having the same color as prohibitory signs can be removed (Fig. 4). After the refinement step, the shape matching is applied to detect the triangular and circular traffic signs.

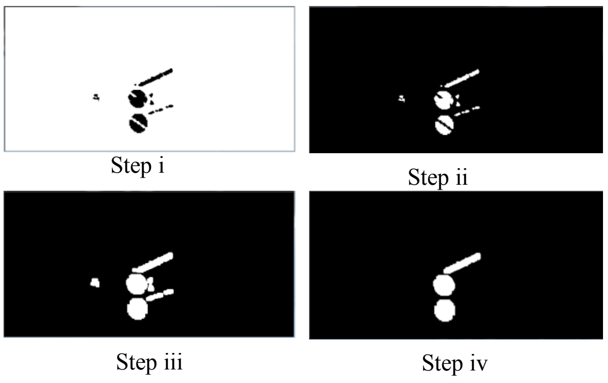


Fig. 4. Demonstration of the refinement steps

3.2.3 Shape matching

In this step, the method proposed by P. Rosin [5] and thresholding method for area of the objects are combined to detect the triangular and circular traffic signs. According to P. Rosin's method, shape estimation is determined by using Affine Moment Invariant calculated by the following equation:

$$I = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \quad (1)$$

where μ is the central moment computed by

$$\mu_{pq} = \sum_{(x,y)\in\Omega_C} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \quad (2)$$

$$f(x,y) = \begin{cases} 1 \; if \; (x,y) \in R_c \\ 0 \quad otherwise \end{cases} \quad (3)$$

There are two metrics to measure ellipticity and triangularity, namely *E* and *T*, respectively, given by:

$$E = \begin{cases} 16\pi^2 I \; if \; I < \dfrac{1}{16\pi^2} \\ \dfrac{1}{16\pi^2 I} \quad otherwise \end{cases} \quad (4)$$

$$T = \begin{cases} 108I \; if \; I < \dfrac{1}{108} \\ \dfrac{1}{108I} \quad otherwise \end{cases} \quad (5)$$

The value *E* = 1 implies s a perfectly circular shape; and the value *T* = 1 indicates a perfect triangular shape.

Threshold values for *E* and *T* are applied to detect candidate objects for triangular and circular traffic signs. Then, the candidates are verified by using an area metric *A* of a candidate object given by

$$A = \begin{cases} \dfrac{w \times h/2}{Area} & \text{if candidate object is triangular} \\ \dfrac{\pi \times r^2}{Area} & \text{if candidate object is circular} \end{cases} \tag{6}$$

Where *w*, *h*, and *r* are width, height, and radius of the candidate object, respectively. *Area* is a number of pixels belonging to the candidate object.

The Table 4 shows the summary of thresholding method to detect triangular and circular traffic signs.

Table 4. Threshold values for detecting triangular and circular traffic signs

Shape	E	T	A
Triangular traffic sign	E< 0.78	0.91< *T*< 1.0	0.95 < A < 1.05
Circular traffic sign	0.98 < E < 1.0	T > 0.68	0.95 < A < 1.05

3.3 Recognizing

In this stage, the candidate objects are first classified into 4 categories: prohibitory signs 1 (with blue background), prohibitory signs 2 (with white background), warning signs, and guide signs as shown in Table 5.

Table 5. Four categories of traffic signs

Attributes	Prohibitory signs 1	Prohibitory signs 2	Warning signs	Guide signs
Color segmentation	red	red	red	blue
Shape	round shape	round shape	round shape	round shape
Background	blue	white	-	-

Next, images of the candidate objects are converted into binary images. To do that, color segmentation and thresholding method are applied for images of the candidate objects as described in Table 6. Fig. 5 shows the result of the binary images.

Table 6. Methods for converting images of the candidate objects to binary images

Categories	Method
Prohibitory signs 1	Apply blue color segmentation. Blue pixels convert to white ones, others convert to black one.
Prohibitory signs 2	Convert the image to gray scale. Convert the image to binary based on a threshold for pixels inside the object.
Warning signs	Convert the image to gray scale. Convert the image to binary based on a threshold for pixels inside the object.
Guide signs	Apply blue color segmentation. Blue pixels convert to black ones, others convert to white ones.

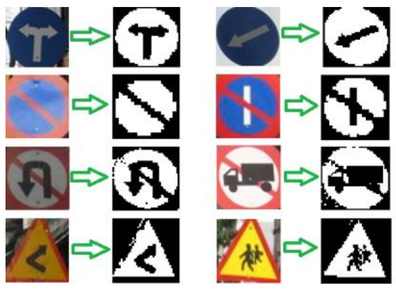


Fig. 5. The result of binary image conversion

Finally, the feature vectors are extracted, and SVM is used to classify each sign of the categories. The processing steps to extract feature vectors are as follows:

- Resize binary image to 40x40
- Convert 40x40 matrix to a column vector 1600x1
- Put the column vector to SVM for classifying the types of traffic signs.

In order to train the SVM, a training set which contains 50 most popular types of traffic signs in Vietnam including 20 prohibitory signs, 20 warning signs, and 10 guide signs is first selected. Then, 2000 feature vectors are created from 2000 samples of binary images of the training set. The labels for all feature vectors are created to indicate their types of signs. OpenCV functions from the class CvSVM are used to perform SVM training and classifying.

4 IMPROVEMENT FOR THE ACCURACY

In order to increase the accuracy of the TSDR system, three simple but effective techniques are proposed.

First, a tracking method for detected traffic signs is applied. When a traffic sign is detected and recognized, this sign is tracked in next frames. In that case, if the sign is detected and recognized as the same type in next three frames, the system will conclude that the sign is presented. This technique can reduce the rate of false detection (detecting a non-traffic sign object) or false recognition (classifying wrong type of traffic sign).

Second, a tracked traffic sign must move backward, because it is assumed that the car goes forward. Therefore, if a candidate object does not move inversely with the car, it may conclude to be a noise object. For example, if the helmet of a motor-rider may have similar features as a traffic sign, the system performs wrong detection.

Third, some advertisement boards or helmets are similar to traffic signs as shown in Fig. 6. They may have round shape, red or blue color, and thus the TSDR system may detect them as traffic sign objects. In order to solve this problem, the images of similar objects appearing on streets are collected as many as possible, and then they are used for training the SVM. By this way the SVM can recognize these non-traffic sign objects.

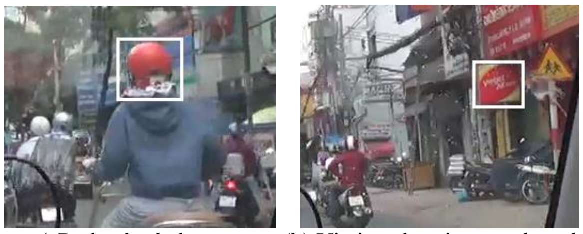

a) Red color helmet (b) Vietjet advertisement board

Fig 6. Some samples of noise objects

5 Multi-Thread Programming For The TSDR System

The proposed algorithm is implemented on an embedded ARM board to create a compact and portable system. The performance of quad-core ARM Cortex-A9 on the board is not powerful than a PC. Therefore, the capability of the ARM processor must be 100% utilized to achieve real-time processing by using multi-thread programming. The multi-thread programming technique is applied in all three stages of the TSDR algorithm.

In the pre-processing stage, color space conversion, white balancing, and de-noising are performed. These steps are repeated for every input frame, thus they require much processing time. To reduce processing time, the input frame is divided into 4 parts which can be processed by 4 threads simultaneously. Each thread can perform color space conversion, white balancing, and de-noising independently by the quad-core ARM processor.

In the detecting stage, the color segmentation is applied to detect candidate regions which can contain traffic signs. Then, each candidate object is processed by shape matching and then sent to the SVM to recognize the type of the traffic sign. In order to enhance real-time capability, each candidate object is assigned for each thread to perform shape matching algorithm.

In the recognizing stage, each candidate object is continued to be processed by each thread to classify for recognition. The number of candidate objects can be more than 4. However, the number of threads is always kept as 4, since the quad-core ARM Cortex-A9 processor is able to process 4 threads at the same time.

In order to implement multi-thread programming, QThread, QMutex, and QSemaphore classes of Qt library [12] are utilized. The QThread provides a platform-independent way to manage threads. A QThread object manages one thread of control within the program. QThreads begin executing in the function named *run()*. By default, this function starts the event loop by calling the other function named *exec()* which will execute a Qt event loop inside the thread. In order to synchronize threads, QMutex and QSemaphore classes are utilized. QMutex provides a means of protecting a variable or a piece of code so that only one thread can access it at a time. QSemaphore is another generalization of mutexes, which can be used to guard a certain number of identical resources.

6 Experimental Results

In our experiment, the test process for the TSDR system including three procedures: hardware testing, software testing, and system testing is performed.

In the hardware testing procedure, ARM processor and all peripherals of the TSDR system are checked. The ARM processor can boot correctly with Linux OS. The USB camera captures images and transfer data to the processor successfully. LCD can display the GUI of the TSDR software. The touch screen works well. The system can output the audio messages about the traffic signs.

In the software testing procedure, the Linux OS, the device drivers, and the TSDR software functions are checked. As a result, the TSDR software runs correctly on ARM board with Linux OS as shown in Fig. 7.

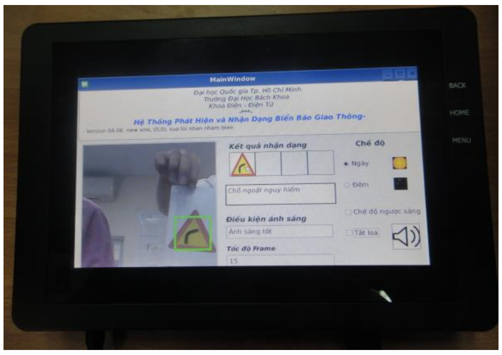


Fig. 7. Functional test for the TSDR system

Finally, in the whole system testing procedure, the TSDR system is tested on the roads for evaluating accuracy and real-time ability.

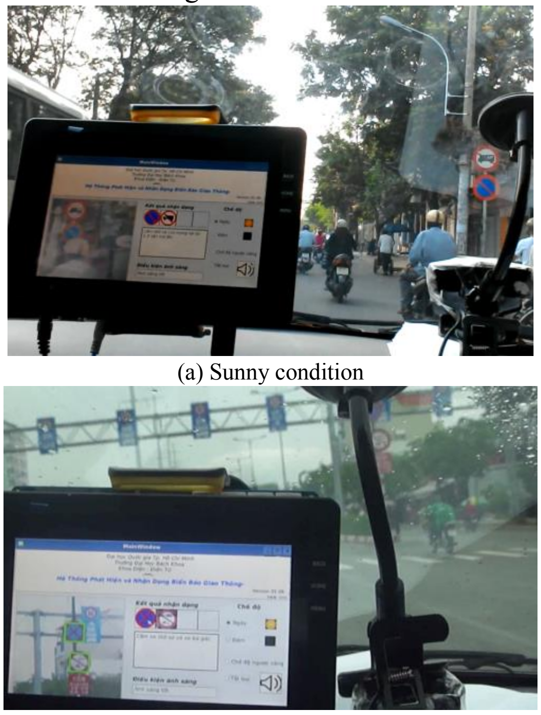
For the accuracy evaluation, the algorithm is first tested on PC with a set of 2500 test images with 50 types of traffic signs. As the result, the TSDR algorithm can detect and recognize the traffic signs at the accuracy of 93% as shown in Table 7. However, this evaluation is only simulation on PC with static image. The TSDR system must be tested on the streets with different conditions of light, weather, occlusion, and disturbed objects to evaluate its performance.

Table 7. Accuracy test of TSDR algorithm

Processing	Total samples	Successful	Unsuccessful
Detection	2500	96%	3.6% undetected 0.4% false detected
Recognition	2500	97%	3%
Overall	2500	93%	7%

Next, the system is tested on the streets in sunny, rainy, and lowlight conditions. To perform the test, the TSDR is mounted on a car behind the front windshield. The test has been done on 20 roads in Ho Chi Minh city. The test result as shown in Table 8 indicates that the accuracy of the TSDR system keeps approximate 93%. Fig. 8 demonstrates some testing results of the TSDR on the streets. More results are posted on the website [13].

In order to understand the reasons for errors of the TSDR system, 3 types of unsuccessful cases are analyzed. The undetected cases are mostly due to the occlusion. The traffic signs can be hidden by trees, advertisement boards, or even the vehicle travelling in front of the camera. This problem can be solved by capturing several frames to confirm the appearance of the traffic signs. The false detected cases occur when objects which are similar to traffic signs appear. The number of these cases can be reduced by the way that the similar objects are collected and used for training process of SVM classification. This method has been mentioned in the Section 4. The last unsuccessful cases are errors of classification process. These errors are due to very low light conditions, deformed traffic signs, or partial occluded traffic signs.



(a) Sunny condition

(b) Rainy condition

Fig 8. TSDR system testing results on the streets

Table 8. Accuracy test of TSDR system

No.	Street names	No. of signs	Success
1	Lý Thường Kiệt	16	15
2	Hoàng Văn Thụ	4	4
3	Lê Văn Sỹ	16	14
4	Võ Thị Sáu	12	12
5	Ba Tháng Hai	8	8
6	Điện Biên Phủ	29	26
7	Nguyễn Đình Chiểu	19	19
8	Bà Huyện Thanh Quan	14	12
9	Lý Thái Tổ	10	9
10	Thành Thái	14	13
11	Phạm Văn Đồng	22	20
12	Võ Văn Kiệt	12	11
13	Nguyễn Văn Cừ	7	6
14	Bàu Cát	10	10
15	Âu Cơ	8	8
16	Bình Thới	12	11
17	Hàn Hải Nguyên	9	9
18	Hoàng Sa	5	5
19	Cách Mạng Tháng 8	14	12
20	Lê Lợi	19	17
TOTAL		260	241
Success rate: 241/260 = 0.927			

For the analysis of real-time capability, the simulation is first examined on a PC with Intel core-i5 2.5GHz processor. The proposed algorithm is easily to adapt the throughput of 20-25fps. Next, the real-time capability of the implementation of TSDR is examined on Friendly ARM board with quad-core ARM Cortex-A9 processor. The performance of the system without multi-threading technique is about 10-11fps. After the multi-threading technique has been applied for parallel processing, the throughput is 15-18fps.

The proposed system is compared with three other works. The first reference work proposed by Thanh Bui-Minh et al [6] is based on color segmentation and SVM classification. The set of traffic sign images is from Irish and UK. The second work presented by Jack Greenhalgh et al [9] used maximally stable external regions for detection and SVM with HOG features for classification. The last reference offered by Chi-Yi Tsai et al [14] utilized centroid-to-contour (CtC) distances for road sign detection and SVM for classification.

Table 9 shows the detail comparison of our proposed method and others. The works of [6] and [9] are tested on PC, and thus they have been not proved for practical embedded applications on portable devices. The works of [14] and our algorithm are verified on the embedded ARM boards which are able to equip on cars. In the work [14], the processing rates of the system are 22fps and 30fps for the Radxa Rock Pro and ASUS PF500KL embedded platforms, respectively. The accuracy can achieve up to 99%. However, the authors only detect and recognize 10 types of speed-limit traffic signs. The experiments were done with the ideal condition of in-house testing database.

The proposed system has some advantages compared to the other works in term of accuracy and processing time. The overall accuracy of the proposed system is higher than [6] and [9], but lower than [14]. However, the work [14] was tested with the ideal condition of in-house testing database of only 10 speed-limit signs. Our system was also verified with in-house conditions, and thus the accuracy rate of the system is very high. Nevertheless, the proposed system has been tested on the real streets in Vietnam with many noise objects which can affect to the accuracy rate of the system. The number of 50 types of traffic signs is enough for a real application. The system is implemented on a cost-effective embedded system ARM board which is portable and easy to be mounted on a car.

Table 9. Comparison between the proposed system and other works

Features	[6]	[9]	[14]	Proposed
Number of sign types	69	127	10 (only speed-limit signs)	50
Test platform	Dual core 2.2GHz PC	Intel core-i5 3.33GHz PC	Radxa Rock Pro and ASUS PF500KL	Intel Core-i5 2.5GHz PC and quad-core ARM Cortex-A9
Real-time	8fps	20fps	22fps and 30fps	20fps on PC 15fps on ARM
Overall Accuracy	86.7%	89.2%	99%	93%

7 CONCLUSION

The design and implementation of TSDR system on the ARM-based embedded board have been presented in this paper. The system was designed following the embedded system process and has been fully tested for hardware, software and the whole system. The proposed TSDR algorithm can achieve the accuracy of 93% at 15fps. The proposed system is portable and ready to be equipped on car to support drivers tracking traffic signs. The system can detect and recognize a set of 50 most popular traffic signs in Vietnam.

REFERENCES

[1]. S. Maldonado-Bascón, J. Acevedo-Rodríguez, A. Fernández-Caballero, and F. López-Ferreras, *An optimization on pictogram identification for the road-sign recognition task using SVMs*, Computer Vision and Image Understanding, vol. 114, no. 3, pp. 373-383, 2009.

[2]. C. Y. Fang, C. S. Fuh, S. W. Chen, and P. S. Yen, *A Road Sign Recognition System Based on Dynamic Visual Model*, Proceedings on Computer Vision and Pattern Recognition, 2003.

[3]. H. Fleyeh, "Traffic and road sign recognition", Dalarna University,Sweden, 2008

[4]. Luis David Lopez and Olac Fuentes, "Color-Based Road Sign Detection and Tracking", International Conference on Image Analysis and Recognition (ICIAR), Montreal, CA, August 2007.

[5]. P. Rosin, "Measuring shape: Ellipticity, rectangularity, and triangularity", Machine Vision and Applications, vol. 14, no. 3, pp. 172-184, 2003.

[6]. Thanh Bui-Minh, Ovidiu Ghita, Paul F. Whelan, and Trang Hoang, "A Robust Algorithm for Detection and Classification of Traffic Signs in Video Data", ICCAIS Saigon Vietnam, 2012.

[7]. C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, T. Koehler, "A System for Traffic Sign Detection,

Tracking, and Recognition Using Color, Shape, and Motion Information", Proceedings. IEEE Intelligent Vehicles Symposium, 2005.

[8]. Fistrek, T.; Loncaric, S., "Traffic sign detection and recognition using neural networks and histogram based selection of segmentation method", Processdings ELMAR, 2011.

[9]. Jack Greenhalgh and Majid Mirmehdi, "Real-Time Detection and Recognition of Road Traffic Signs", IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, December 2012.

[10]. Rong-Qiang Qian, Yong Yue, Frans Coenen, Bai-Ling Zhang, "Traffic Sign Recognition Using Visual Attribute Learning And Convolutional Neural Network", Proceedings of the International Conference on Machine Learning and Cybernetics, 2016.

[11]. A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features", Pattern Recognition, vol. 43, no. 1, pp. 416-430, 2010.

[12]. Jasmin Blanchette ; Mark Summerfield, C++ GUI Programming with Qt 4, Prentice Hall, 2008.

[13]. The TSDR system testing results, http://www4.hcmut.edu.vn/~tqvinh/traffic-sign-recognition.html.

[14]. Chi-Yi Tsai, Hsien-Chen Liao, Kuang-Jui Hsu, "Real-time embedded implementation of robust speed-limit sign recognition using a novel centroid-to-contour description method", IET journals, Vol. 11 Iss. 6, pp. 407-414, 2017.

Dr. Truong Quang Vinh received the B.E. degree from Ho Chi Minh University of Technology, Vietnam, in 1999, the M.E. degree in Computer Science from the Asian Institute of Technologies, Bangkok, Thailand, in 2003; and Ph.D. degree in Computer Engineering at Chonnam National University, Korea, in 2010. Currently, he is a lecturer in Department of Electrical and Electronics Engineering, HCM University of Technology. His research interests include VLSI design on FGPA, ASIC design, embedded system/system-on-chip design, and image processing.

Thiết kế và hiện thực hệ thống nhúng di động cho phát hiện và nhận dạng biển báo giao thông thời gian thực

Trương Quang Vinh*

Trường Đại học Bách khoa, ĐHQG-HCM
*Tác giả liên hệ: tqvinh@hcmut.edu.vn

Ngày nhận bản thảo: 06-5-2018; Ngày chấp nhận đăng: 17-9-2018; Ngày đăng: 30-12-2018

Tóm tắt—Ngày nay, hệ thống hỗ trợ lái xe được nhiều công ty quan tâm để sản xuất những ôtô hiện đại. Trong hệ thống đó, giải thuật phát hiện và nhận diện biển báo giao thông là một vấn đề thách thức mà các nhà nghiên cứu cố gắng giải quyết. Bài báo này trình bày thiết kế và hiện thực hệ thống phát hiện và nhận dạng biển báo giao thông thời gian thực (TSDR – Traffic Sign Detection and Recognition) để giúp người tài xế nhận biết các biển báo giao thông trên đường phố. Các đặc tính chính của hệ thống TSDR là khả năng xử lý thời gian thực và độ chính xác cao. Để đạt được các mục tiêu này, một phương pháp kết hợp các kỹ thuật tiên tiến được đề nghị bao gồm phân chia màu sắc thích nghi, so khớp hình dạng, và máy vectơ hỗ trợ (SVM). Bên cạnh đó, một kỹ thuật lập trình đa luồng được áp dụng để nâng cao khả năng xử lý thời gian thực của hệ thống. Hệ thống TSDR được thực hiện trên một bo mạch hệ thống nhúng di động với bộ xử lý ARM Cortex-A9. Hệ thống TSDR đã được kiểm tra trên đường phố thành phố Hồ Chí Minh. Các thí nghiệm cho thấy hệ thống đề nghị có thể phát hiện và nhận diện các biển báo giao thông với độ chính xác 93% ở tốc độ 15 khung hình / giây.

Từ khóa—**Biển báo giao thông, phân đoạn màu, khớp hình dạng, máy vectơ hỗ trợ, vi xử lý ARM**