Traffic Sign Recognition Using Neural Network on OpenCV: Toward Intelligent Vehicle/Driver Assistance System

Auranuch Lorsakul[1] and Jackrit Suthakorn[*,2]

Center for Biomedical and Robotics Technology (BART LAB),
Department of Biomedical Engineering, Faculty of Engineering,
Mahidol University
25/25 Putthamonthon 4 Road, Salaya,
Putthamonthon, Nakornpathom, 73170
THAILAND

Phone: 662-8892138 ext. 6446
Fax: 662-4414250
[1]auranuch@bartlab.org, [2]egjst@mahidol.ac.th
*Corresponding Author
www.bartlab.org

Abstract

Traffic Sign Recognition (TSR) is used to regulate traffic signs, warn a driver, and command or prohibit certain actions. Fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver and significantly increase driving safety and comfort. Automatic recognition of traffic signs is also important for an automated intelligent driving vehicle or for driver assistance systems. This paper presents a study to recognize traffic sign patterns using Neural Network technique. Images are pre-processed with several image processing techniques, such as, threshold techniques, Gaussian filter, Canny edge detection, Contour and Fit Ellipse. Then, the Neural Networks stages are performed to recognize the traffic sign patterns. The system is trained and validated to find the best network architecture. The experimental results show highly accurate classifications of traffic sign patterns with complex background images as well as the results accomplish in reducing the computational cost of this proposed method.

Keywords *- Traffic Sign Recognition, Intelligence Vehicle, Neural Network*

1. Introduction

In traffic environments, Traffic Sign Recognition (TSR) is used to regulate traffic signs, warn drivers, and command or prohibit certain actions. Fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver, and thus, significantly increase driving safety and comfort. Generally, traffic signs provide the driver with a variety of information for safe and efficient navigation. Automatic recognition of traffic signs is, therefore, important for automated intelligent driving vehicle or for driver assistance system. However, identification of traffic signs with respect to various natural background viewing conditions still remains a challenging task. Traffic Sign Recognition Systems usually have been developed into two specific phases [1-7]. The first phase is normally related to the detection of traffic signs in a video sequence or an image using image processing. The second one is related to recognition of those detected signs, which deals with the interest of performance in an artificial neural network. The detection algorithms are normally based on shape or color segmentation. The segmented potential regions are extracted as input in recognition stage. The efficiency and speed of the detection play important roles in the system. To recognize traffic signs, various methods for automatic traffic sign identification have been developed and shown promising results. Neural Networks precisely represents a technology used in traffic sign recognition [1-8]. One specific area in which many neural network applications have been developed is the automatic recognition of signs. Fig. 1 illustrates the concept of automated intelligent driving vehicle or for driver assistance system.

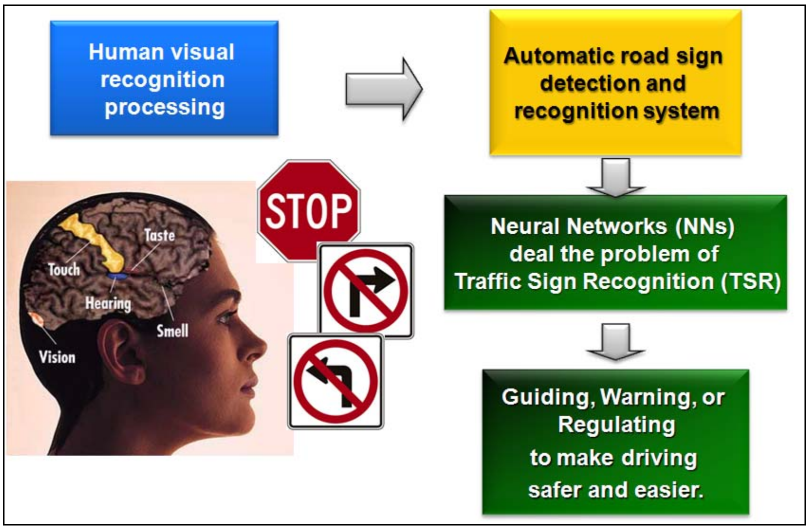


Fig. 1 The concept of automated intelligent driving vehicle or for driver assistance system

The difficulties of traffic sign detection and recognition are involved with the performance of a system in real time. Highly efficient algorithms and powerful performance hardware are required in the system [3]. Furthermore, the environmental constraints include lighting, shadow occlusion, air pollution, weather conditions (sunny, rainy, foggy, etc.) as well as additional image distortions, such as, motion blur, vehicle vibration, and abrupt contrast changes which can possibly occur frequently in an actual system [3,7,8].

In recent studies, the detection and recognition of traffic signs have been developed in many research centers. A vision system for the traffic sign recognition and integrated autonomous vehicle was developed as part of the European research project PROMETHEUS at DAIMLER-BENZ Research Center [3]. Moreover, many techniques have been developed for road sign recognition. For example, Pacheco et al. [9] used special color barcodes under road signs for detecting road signs in a vision-based system, however, this took a lot of time and resources. A genetic algorithm was also proposed by Aoyagi and Askura [10] to identify road sign from gray-level images. Because of a limitation of crossover, mutation operator, and optimal solution, it is not guaranteed to achieve results. Color indexing was proposed by Lalonde and Li [11] to approach identifying road sign, unfortunately, the computation time was not accepted in complex traffic scenes.

This paper is proposed to develop the real implementation using in intelligent vehicle. The main objective is to reduce the search space and indicate only potential regions for increasing the efficiency and speed of the system. A higher robust and faster intelligent algorithm is required to provide the necessary accuracy in recognition of traffic signs. In the detection phase, the acquisition image is preprocessed, enhanced, and segmented according to the sign properties of color and shape. The traffic sign images are investigated to detect potential pixel regions which could be recognized as possible road signs from the complex background. The potential objects are then normalized to a specified size, and input to recognition phase. This study investigates only circle and hexagonal shaped objects because these shapes are normally present in many types of traffic signs. Multilayer Perceptron (MLP) with respect to a back propagation learning algorithm is an alternative method to approach the problem of recognizing signs in this work. The image processing tool used in this paper is a free- and non-commercial Intel® Open Source Computer Vision Library (OpenCV) [12]. The remainder of the paper is organized as follows: Section 2 presents a system description of image processing stages to extract the potential feature. The description of Neural Network recognition stage is presented in Section 3. Section 4 and 5 show the representative experimental results/analysis and conclusion, respectively.

2. System Description of Image Processing Stages

2.1 System Overview

The first section is Image Extraction and Sign Detection and Extraction parts. The video images have been taken by a video camera, and the Image Extraction block is responsible for creating images. The Sign Detection and Extraction Stage extracts all the traffic signs contained in each image and generates the small images called blobs. Each blob will be performed by a Form Recognition Stage to be valuable parameter input to Artificial Neural Networks in the Recognition Stage which is the final part. Fig. 2 illustrates the system configuration. Then, the output of traffic sign recognition will be presented. The system is based on previous work presented by [1], [2]. The description of the traffic sign recognition system can be explained as the Traffic Sign Pre-processing Stage and Recognition Core. For the Traffic Sign Pre-processing Stage, it is divided in two parts: Sign Detection and Extraction and Form Recognition Stage. Fig. 3 displays block diagram of the system description.

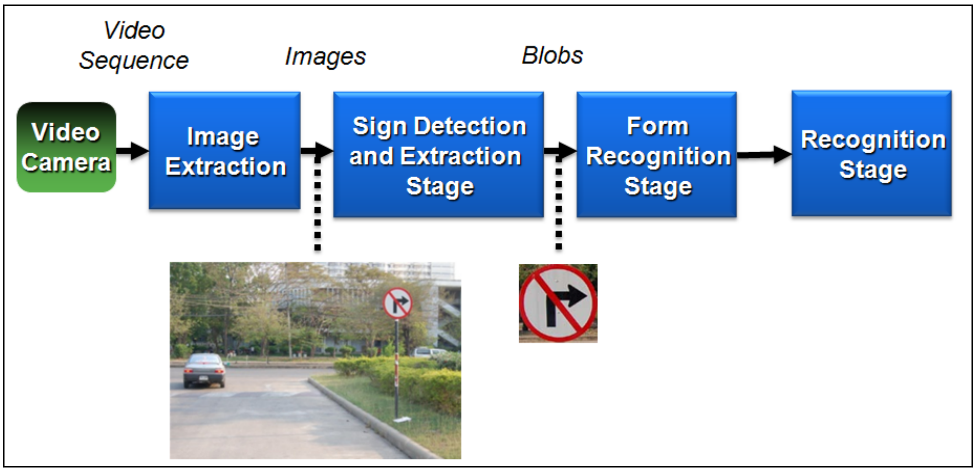


Fig. 2. System configuration

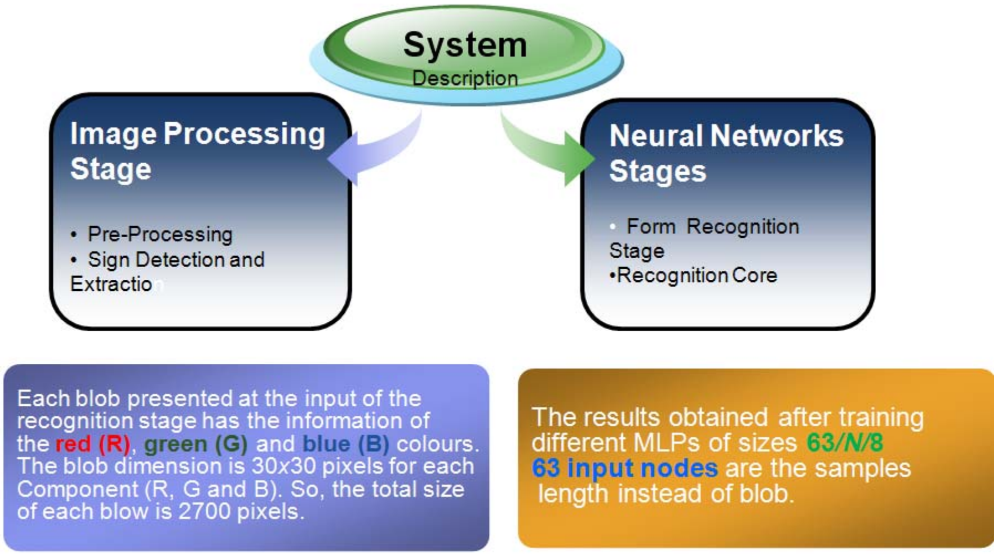


Fig. 3 Block diagram of the system description.

2.2 Traffic Sign Pre-Processing Stage

Sign Detection and Extraction: This stage is the image processing procedure. Image input from the video sequence with natural background viewing image is fed into the system. The image data is read both in color, and black and white mode (Fig.4 (a) and (b) show color and black & white images, respectively).

(a)

(b)

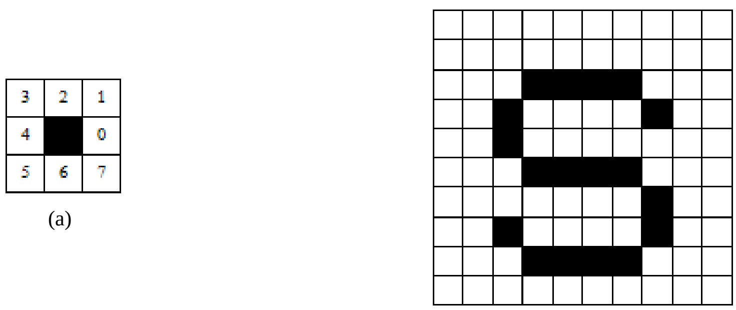
(c)

(d)

Fig. 4 (a) Color Mode Image, (b) Black and White Mode Images, (c) Smooth image using Gaussian filter, and (d) Binary result image after using Gaussian filter and Canny edge detection techniques.

As the black and white mode, image is the base image used to find the threshold of this image. This threshold is the criterion to change an image from black and white to a binary image. Moreover, the binary image is used to find contours and the interesting region later on. Before the black and white image is changed to binary, the technique of smooth image with Gaussians filter and Canny edge detection to enhance image as shown in Fig.4 (c) and 2 (d). Therefore, it shows that the smooth technique has potential to enhance the image to obtain the required region.

According to obtaining the binary image, it is processed to retrieve contours by *find the contour function* for a binary image, and to return the number of retrieved contours which are stored in the chain format. The OpenCV library uses two methods to represent contours. The first method is called the Freeman method or the chain code. For any pixel, all its neighbors with numbers from 0 to 7 can be enumerated as Fig. 5(a). The 0-neighbor denotes the pixel on the right side, etc. As a sequence of 8-connected points, the border can be stored as the coordinates of the initial point, followed by codes (from 0 to 7) that specify the location of the next point relative to the current one. (Fig. 5 (b) illustrates example of Freeman coding of connected components.)



(a)

Chain code : 3 4 4 4 5 6 7 0 0 0 7 6 5 4 4 4 3

(b)

Fig. 5 (a) Contour Representation in Freeman Method, (b) Freeman Coding of Connected Components (Courtesy [12])

Then, the output from the *find contour function* is fed to calculate the ellipse that fits best (in least-squares sense) to a set of 2D points. The OpenCV function used in this task is ‘cvFitEllipse’, a basic task in pattern recognition and computer vision. The sizes of two ellipse axes and point of the center of the ellipse which are the output of this function are used to find the exact sign circle in the image. Fig. 6 displays the structure of the predicted line from the Fit Ellipse function.

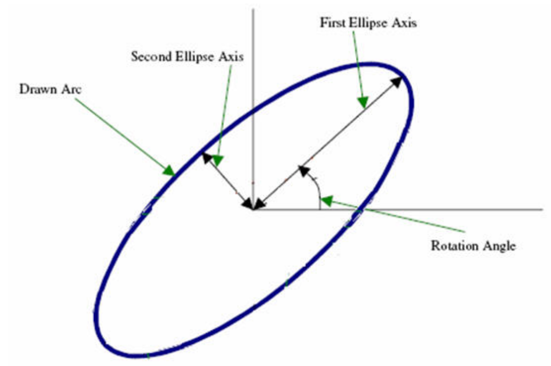


Fig. 6. The structure of predicted line from Fit Ellipse function (Courtesy [12])

The criteria sizes of two ellipse axes are the parameters to decide the sign area in an image. In the program, it checks if vertical and horizontal sizes are not different for more than the criteria of pixels. If the sizes of two ellipse axes pass the criteria that it is a circle of a traffic sign, then the point of the center of the ellipse is used to be a center of the crop image. The algorithm of the crop image is designed according to the structure of the image in pixels. There is an incremental pixel parameter to be set in order to cover all areas of traffic signs.

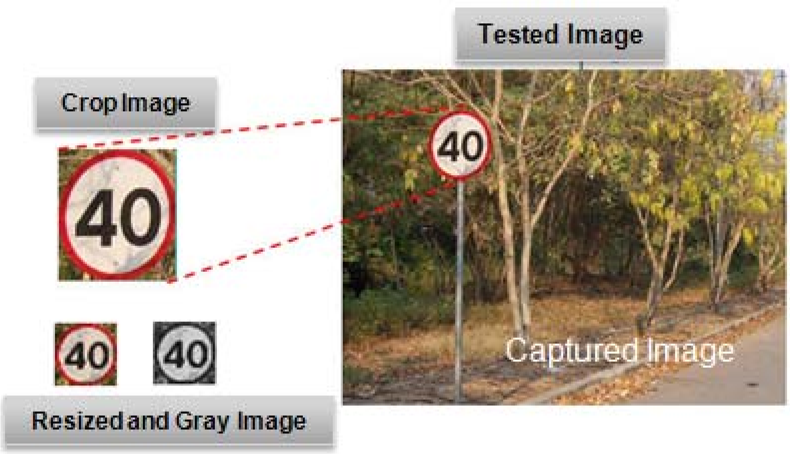


Fig. 7. Crop image from traffic sign region

Furthermore, the crop image is resized to 30x30 pixels in each RGB layer, so that there are totally 2,700 pixels for each resized region. The crop region is converted from three channels (R, G, and B) to a single channel of grayscale subsequently. The single layer gray scale image is called a “blob” which presented as input of the Neural Network. The procedure to extract this image information to be parameters of the input layer of MLP is described in next section. Fig. 7 shows a crop image from the traffic sign region, and Fig. 8 presents the whole pre-process stage.

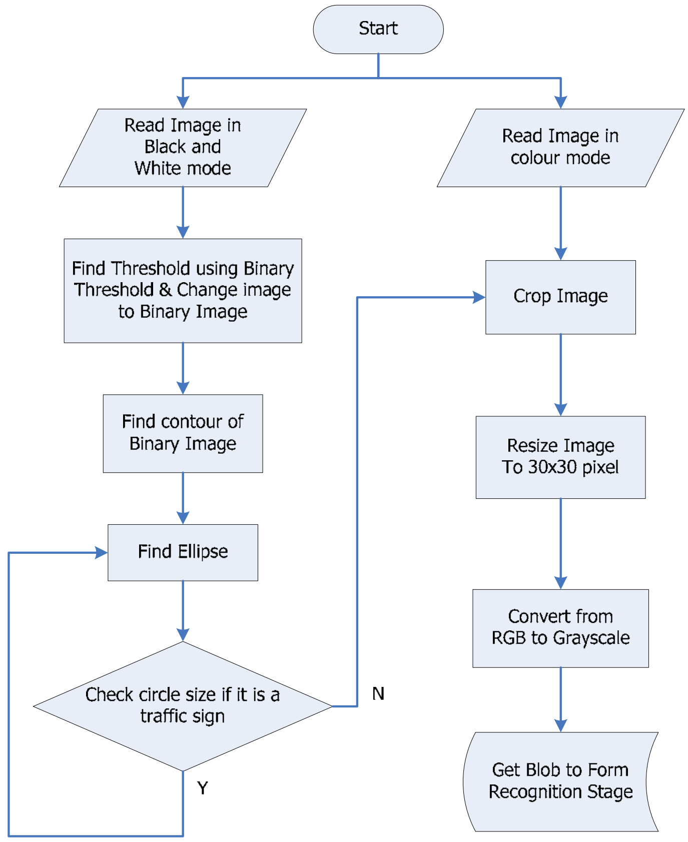


Fig. 8. Block diagram of pre-processing stage

3. System Description of Neural Networks Stages

3.1 Form Recognition Stage

In this stage the blob that represents the input to Neural Networks will be preprocessed to obtain the information of the red (R), green (G) and blue (B) colors. The preprocess stage reduces the number of MLP input by using blob. The dimension of the blob is 30x30 pixels for each component (R, G and B). Therefore, the total size is 2700 pixels. The total input to MLP is 63 nodes. They consists of 3 normalized average maximum pixel values, MR, MG and MB, 30 inputs from a vertical histogram (vh) and 30 inputs from a horizontal histogram (hh) [1], [2].

First, in each blob, the essential matrix **B** is formed. Matrix **B** is considered as matrix that contains the three color components of the blob. After that matrix **B'** is created to be the results of representing **B** in a grey scale. The conversion from RGB format to grey scale format is calculated with Eq. (1). The element values of $b_{i,j}$ and $b'_{i,j}$ represent the *i-th* row and *j-th* column of the matrix **B** and **B'**, respectively, where both indexes (*i* and *j*) varies from 1 to 30.

$$b'_{i,j} = 0.49 b_{i,j} + 0.29 b_{i+30,j} + 0.22 b_{i+60,j} \tag{1}$$

Next, form the 3 normalized average maximum pixel parameters. The normalized averages to the maximum pixel value (28) of R (MR), G (MG) and B (MB) are calculated to be the additional input to MLP. These values are between 0 and 1. The components can be calculated with Eq. (2), (3) and (4), respectively.

$$MR = \frac{1}{256}\left(\frac{1}{900}\sum_{i=1}^{30}\sum_{j=1}^{30} b_{i,j}\right) \tag{2}$$

$$MG = \frac{1}{256}\left(\frac{1}{900}\sum_{i=31}^{60}\sum_{j=1}^{30} b_{i,j}\right) \tag{3}$$

$$MB = \frac{1}{256}\left(\frac{1}{900}\sum_{i=61}^{90}\sum_{j=1}^{30} b_{i,j}\right) \tag{4}$$

Finally, the 30 vertical parameters (vh) and 30 horizontal parameters (hh) are calculated with Eq. (5) and (6), respectively.

$$vh_i = \frac{1}{30}\sum_{j=1}^{30}(b'_{i,j} > T) \quad , \; i = 1, 2, ..., 30 \tag{5}$$

$$vh_i = \frac{1}{30}\sum_{i=1}^{30}(b'_{i,j} > T) \quad , \; j = 1, 2, ..., 30 \tag{6}$$

where $T = \frac{1}{900}\sum_{i=1}^{30}\sum_{j=1}^{30} b'_{i,j}$, adaptive threshold.

Thus the calculated totally 63 parameters, MR, MG, MB, 30 inputs from a vertical histogram (vh) and 30 inputs from a horizontal histogram (hh) are provided into each node in neural network architecture, MLP.

3.2 Traffic Sign Data Base Description

The database has been divided into training, validation and test sets. The first one is used to train the MLP. The second one is used as a validation set during the MLP training to improve generalization property of the network. The last one is used to evaluate the performance of the trained MLP. To select appropriated network architecture for both generalization and approximation abilities, the number of layers and the number of hidden neurons per layer should be determined. Basically, using a single hidden layer tends to interact with global problems. Therefore, in this paper, the single layer perceptron is employed together with the verification of the number of hidden nodes. To find the number of hidden neurons for the appropriate network architecture, the use of a cross-validation is approached. The method is to partition the data set into a training set, $\mathbf{T}_{training}$ and a test set, $\mathbf{T}_{test}$. Then subdivide $\mathbf{T}_{learning}$ into two subsets: one to train the network, $\mathbf{T}_{learning}$, and the other to validate the network, $\mathbf{T}_{validation}$. We train different network architectures on $\mathbf{T}_{learning}$ and evaluate their performance on $\mathbf{T}_{validation}$. The best network is selected, and then finally retrains this network architecture on $\mathbf{T}_{training}$. The test set, $\mathbf{T}_{test}$, is utilized to test for generation ability of network [15]. The training set consists of two main data. The first set combines 4 patterns of sign images without problems. In each sign pattern, there are 5 different views of the same sign board. The other set combines the same 4 patterns of sign images with 5 different distortion problems. The distortions are stain color, noise, pixilated distortion, blur, and color distortion. Fig. 9 shows the training set, $\mathbf{T}_{training}$, (a) illustrates the 4 patterns of training data without problem in 5 different views, and (b) illustrates the 4 patterns of training data with 5 different distortions problems (stain color, noise, pixilated distortion, blur, and color distortion). These two sets are provided simultaneously to train the network. Weights of the network have been adjusted according to the training data set. Then, the validation set, $\mathbf{T}_{validation}$, verifies the network by observing to Least-

Square-Criterion and Kullback-Leibler Distance measurements. Fig. 10 shows the $\mathbf{T}_{validation}$ data which are different images from those in the training set.

(a)

(b)

Fig. 9. (a) The 4 patterns of training data without problem in 5 different views, and (b) illustrates the 4 patterns of training data with 5 different distortions problems (stain color, noise, pixilated distortion, blur, and color distortion).

Fig. 10. Validating data which are different images from those in the training set.

3.3 Recognition Core

Pattern recognition can be implemented by using a trained feed-forward neural network. Multilayer neural network is one of the most new discovery and widespread disseminated effective methods [13]. The supervised backpropagation which is frequently used in many applications is a learning technique used in this paper. In this research, multilayered feedforward neural network learns by using a backpropagation algorithm. Because of its easy implementation, fast and efficient operation, multilayered feedforward network is widely accepted for applying to such this application [15]. In the training process, the network obtains 63 essential training data parameters which are extracted by the Form Recognition Stage. The weights of 63 node-input layer are altered, and then the parameters are passed to the hidden layer respectively. When performing the network, the network recognizes the input patterns and attempts to distinguish the associated output patterns. The network progressively tunes the associated outputs with respect to input patterns until approaching the criteria. Subsequently, the best network architecture is accomplished selected by the validation data. After validating, the network finally delivers the output corresponded to a training input pattern with the least distance from those in input pattern. Fig. 11 shows MLP structure with 63 input nodes. The activation function and the least distance measurements performed in this research are described in the following sections. Fig. 12 displays the state of Recognition Core stage.

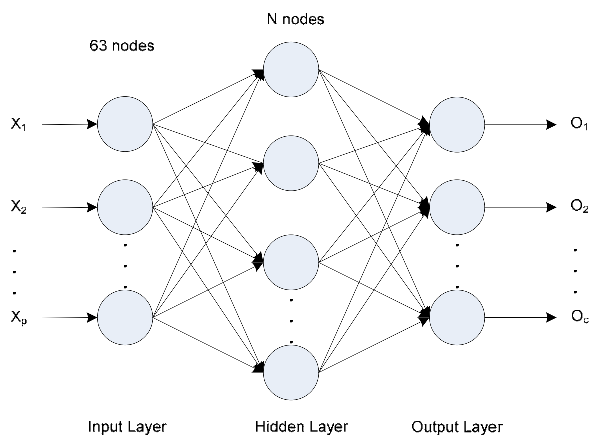


Fig. 11. MLP Structure with 63 input nodes

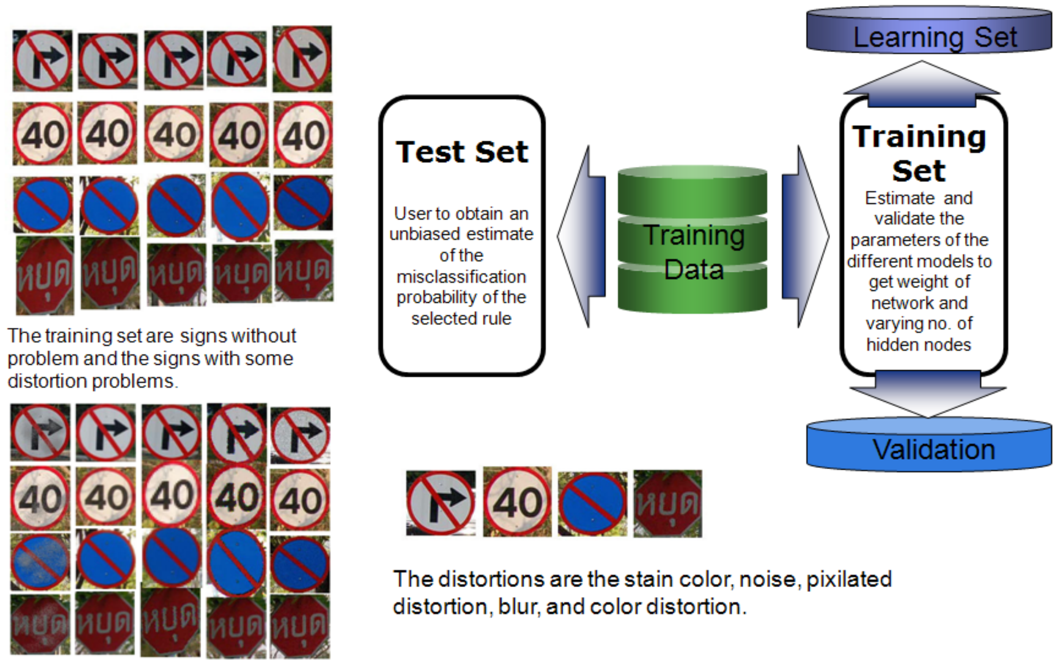


Fig. 12 The state of Recognition Core stage

A. Activation Function

An activation function for a backpropagation should have several important characteristics, such as, continuous, differentiable, and monotonically non-decreasing [13]. The desirable activation function used in this paper is binary sigmoid function, which has a range of (0, 1) and is defined as

$$\delta_j(x_j) = \frac{1}{1+\exp(-\lambda_j x_j)} \tag{7}$$

where λ_i is a gain scale factor. In the limit principle, as $\lambda_i \rightarrow \infty$ the smooth sigmoid function approaches the non-smooth binary threshold function. Fig. 13 illustrates Binary Sigmoid, range (0, 1) with the shifted version of the signal. The sigmoidal function basically contains the very useful mathematical properties, monotonicity and continuity. Monotonicity property means that the function *f(x)* either always increases or always decreases as *x* increases. Continuity property means that there are no breaks in the function: it is smooth. These parameters are intrinsic properties that eventually assist network power to approximate and generalize by learning from data.

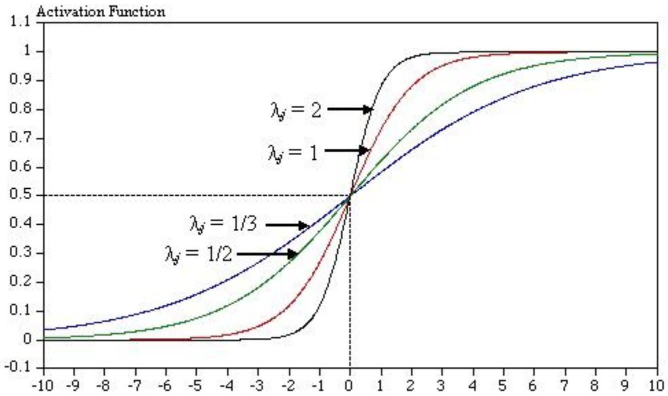


Fig. 13. Binary Sigmoid, range (0, 1)

B. Least-Square-Criterion

In neural net science, estimation is usually based on the minimization of a distance measurement. Typically, in most applications, the least squares criterion is chosen as the distance measurement. A sample of Q individuals with known input $x^{(n)} = (x_1^{(n)}, ..., x_p^{(n)})'$ and known class level $y^{(n)} \in \{0,1\}$ is utilized. Eq. (8) expresses the relationship of the Least-Square-Criterion distance.

$$D_{LS}(w) = \frac{1}{Q}\sum_{q=1}^{Q}\sum_{r=1}^{R}\left(y^{(r)}(q) - f^{(r)}(x(q), w(q))\right)^2 \tag{8}$$

where Q is the number of training patterns, and $f(x^{(n)}, w)$ is output from activation function. w is the weight of MLP.

C. Kullback-Leibler Distance

This is the other measurement criteria selected to use in this work. The Kullback-Leibler distance is an alternative criterion especially for classification problems in neural net science [14]. This parameter together with the Least-Square-Criterion assists to make a decision to select the best network by validation technique. Eq.(9) expresses the relationship of the Kullback-Leibler distance.

$$D_{KL}(w) = -\frac{1}{Q}\sum_{q=1}^{Q}\sum_{r=1}^{R}\log\left[1 - \left|y^{(r)}(q) - f^{(r)}(x(q), w(q))\right|\right] \tag{9}$$

4. Experimental Result and Analysis

To perform the experiment, the training data set, $\mathbf{T}_{training}$, is supplied into the network to update all weights. This data set is used for learning the 4 patterns of the traffic signs. The network is validated the performance by the validating set, $\mathbf{T}_{validation}$. It is activated to find the appropriate number of hidden neurons. The hidden neuron number is varied to various values, and then the network performance is measured by the Least-Square-Criterion and the Kullback-Leibler distance. The result of the number of hidden nodes with respect to the least-square-criterion-error and Kullback-Leibler-error is shown in Table 1. Basically, the experiment is performed in numerous numbers of hidden neurons of the network architecture, but Table 1 shows only the significant results. Regarding to the results, the good architecture starts at the number of hidden neurons of 42. With the least-square-criterion-error, this criteria parameter gradually reduced to the satisfied value which equals 0.0865. While the Kullback-Leibler-error equals 0.3231.

Table 1. Error Analysis with Varied Hidden Neuron Numbers.

Hidden Neurons (N)	**Least- -Square-Criterion –Error**	**Kullback-Leibler Error**
7	0.3078	1.0311
10	0.3426	1.1221
20	0.3089	1.0052
30	0.1309	0.5017
40	0.1554	0.5724
42	0.0865	0.3231
45	0.0712	0.2688
50	0.0556	0.3322
60	0.0356	0.2018
70	0.0374	0.2369
80	0.0247	0.1610

After choosing the appropriate network, the test set, $\mathbf{T}_{test}$, such the examples displayed in Fig. 14 was performed to evaluate the network performance. The test data include traffic signs with the background environment. They have to perform to detect the interested region of a sign board in the image processing stage. The test sign images including the distortion images were provided into the program in order to identify the network generalization and classification abilities. The computational cost of the proposed method has been calculated from the 52 test sign images. The average processing time is 37.27 milliseconds per frame. Fig. 15 shows the plot of computational time of test image processing. The graph shows that the processing times can be separated into two ranges which depend on the complexity of the images. Because the algorithm attempts to detect the circle or ellipse in the test images, this requires more processing time in the much complex background images that have high numbers of the potential area candidates.

Fig. 14 The examples of test images with background used to perform for the network performance evaluation.

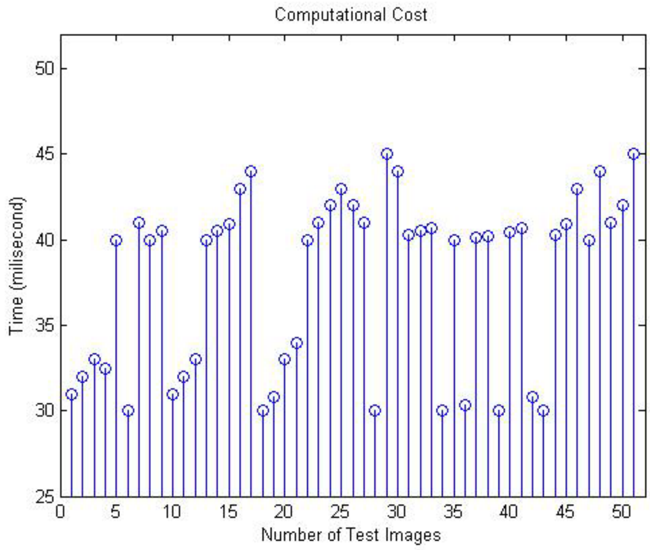


Fig 15. Computational cost plot of the proposed method

5. Conclusion

This study discussed Traffic Sign Recognition (TSR) using Neural Networks technique. The images were pre-processed in stages with image processing techniques, such as, threshold technique, Gaussian filter, Canny edge detection, Contour and Fit Ellipse. The images have been extracted to small area region of traffic sign called "blob." The main reason to select this method is to reduce the computational cost in order to facilitate real time implementation. Then, the Neural Network stages were performed to recognize the traffic sign patterns. The first strategy is to reduce the number of MLP inputs by pre-processing the traffic sign image, and the second strategy is to search for the best network architecture by selecting a suitable error criterion for training.

The system was trained by a training data set, and validated by a validating data set to find the suitable network architecture. The cross-validation technique was implemented by a training data set, validating data set, and test set. The experiments show consistent results together with accurate classifications of traffic sign patterns with complex background images. The processing time in each frame of an image is approached and satisfied to apply in the real intelligent vehicle or driver assistance application.

Acknowledgements

The first Author would like to thank TGIST for partial support scholarship (TG-11-14-49-003M). Authors would like to thank to Prof. Panrasee Ritthipravat, Mahidol University, Thailand, for her valuable suggestions. Authors also would like to thank all members of Center for Biomedical and Robotics Technology (BARTLAB), Faculty of Engineering, Mahidol University, and Faculty of Engineering, Mahidol University to give us opportunity to perform such great research.

References

[1] R. Vicen-Bueno, R. Gil-Pita, M.P. Jarabo-Amores and F. L´opez-Ferreras, "Complexity Reduction in Neural Networks Applied to Traffic Sign Recognition", *Proceedings of the 13th European Signal Processing Conference*, Antalya, Turkey, September 4-8, 2005.

[2] R. Vicen-Bueno, R. Gil-Pita, M. Rosa-Zurera, M. Utrilla-Manso, and F. Lopez-Ferreras, "Multilayer Perceptrons Applied to Traffic Sign Recognition Tasks", *LNCS 3512*, *IWANN 2005, J. Cabestany, A. Prieto, and D.F. Sandoval (Eds.)*, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 865-872.

[3] H. X. Liu, and B. Ran, "Vision-Based Stop Sign Detection and Recognition System for Intelligent Vehicle", *Transportation Research Board (TRB) Annual Meeting 2001*, Washington, D.C., USA, January 7-11, 2001.

[4] H. Fleyeh, and M. Dougherty, "Road And Traffic Sign Detection And Recognition", *Proceedings of the 16th Mini - EURO Conference and 10th Meeting of EWGT*, pp. 644-653.

[5] D. S. Kang, N. C. Griswold, and N. Kehtarnavaz, "An Invariant Traffic Sign Recognition System Based on Sequential Color Processing and Geometrical Transformation", *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*
Volume , Issue , 21-24 Apr 1994, pp. 88 – 93.

[6] M. Rincon, S. Lafuente-Arroyo, and S. Maldonado-Bascon, "Knowledge Modeling for the Traffic Sign Recognition Task", Springer Berlin / Heidelberg Volume 3561/2005, pp. 508-517.

[7] C. Y. Fang, C. S. Fuh, P. S. Yen, S. Cherng, and S. W. Chen, "An Automatic Road Sign Recognition System based on a Computational Model of Human Recognition Processing", *Computer Vision and Image Understanding,* Vol. 96 , Issue 2 (November 2004), pp. 237 – 268.

[8] 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 of the 2005 IEEE Intelligent Vehicles Symposium*, Las Vegas, USA., June 6 - 8, 2005.

[9] L. Pacheco, J. Batlle, X. Cufi, "A new approach to real time traffic sign recognition based on colour information", *Proceedings of the Intelligent Vehicles Symposium*, Paris, 1994, pp. 339–344.

[10] Y. Aoyagi, T. Asakura, “A study on traffic sign recognition in scene image using genetic algorithms and neural networks”, *Proceedings of the IEEE IECON Int. Conf. on Industrial Electronics, Control, and Instrumentation*, Taipei, Taiwan, vol. 3, 1996, pp. 1838–1843.

[11] M. Lalonde, Y. Li, Detection of Road Signs Using Color Indexing, Technical Report CRIM-IT-95/12-49, Centre de Recherche Informatique de Montreal. Available from: <http://www.crim.ca/sbc/english/cime/> publications.html, 1995.

[12] Intel Corporation, “Open Source Computer Vision Library,” Reference Manual, Copyright © 1999-2001, Available: www.developer.intel.com

[13] L. Fausett, “Fundamentals of Neural Networks Architectures, Algorithms, and Applications”, Prentice Hall Upper Saddle River, New Jersey, 1994.

[14] G. Schwarzer, W. Vach, and M. Schumacher, “On the Misuses of Artificial Neural Networks for Prognostic and Diagnostic Classification in Oncology,” *Statistics in Medicine,* 2000, Vol. 19, pp. 541-561.

[15] S. Kumar, “Neural Networks: A Classroom Approach”, Mc Graw Hill 2005.