

# Spatial Feature Extractions to Reduce Intra-Class Variability in Traffic Sign Templates

Hwee Ling Wong<sup>1\*</sup> and Chaw Seng Woo<sup>2</sup>

<sup>1</sup>Faculty of Engineering, Multimedia University, Malaysia; hlwong@mmu.edu.my

<sup>2</sup>Faculty of Computer Science and Information Technology, University of Malaya, Malaysia;

cswoo@um.edu.my

## Abstract

This paper presents shape features and a normalization scheme to address intra-class variation challenge of various traffic sign template designs. Traffic sign design around the world must conform to certain guidelines. However, there may be spatial differences among traffic signs belonging to a same class. In this work, we propose 22 features to discriminate 23 classes of traffic signs. This low number of features would minimize the classification time. We highlight the usage of location and directional information of local shape features. The application of the features is explained using the Naïve Bayes classifier for classification of the images. A dataset containing traffic sign templates with up to a maximum of 22 countries per class was used in our experiment to simulate intra-class variations. The features were compared against twelve other classifiers. The Naive Bayes classification technique performs the best with 99.4% classification accuracy and average of 0.43ms classification time per feature set. We proved that the features proposed are effective to discriminate inter-class images even though there are intra-class differences.

**Keywords:** Feature Extraction, Image, Shape Analysis, Traffic Sign

## 1. Introduction

Traffic signs are road markers placed along the road to provide information of the road conditions ahead. They may appear in the form of warning, prohibitive, mandatory, directional or temporary signage. The design of a traffic sign is regulated by the government body of a country. The design itself usually conforms to certain worldwide guideline, such as the Vienna Convention on Road Signs and Signals<sup>1</sup>. However, discrepancy of the traffic sign design among different countries exists. One reason is many of points in the guideline are given in sentences but not numerically. Secondly, the traffic sign design booklet that is published by a government body allows variations in the actual physical traffic sign<sup>2</sup>. For instance, the width of an arrow in a bend left sign can have an error of  $\pm m$  cm or the angle of a curvature can have an error of  $\pm n^\circ$ . In addition, the construction of the physical traffic sign is usually subcontracted to private contractors. A contractor at time  $t_1$  may be different from

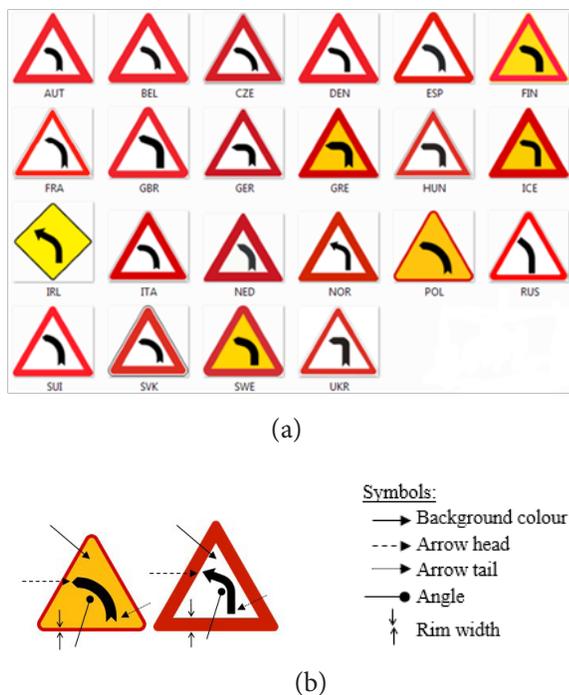
contractor at time  $t_2$  for traffic sign placement at area  $a_1$ . Likewise, traffic sign for area  $a_1$  may have a have a different contractor at area  $a_2$  at time  $t_1$ . Due the difference of the contractors and error allowable in the official traffic sign design handbook, the actual traffic sign with that conveys the same message may be different within the same country as well.

Before the physical traffic sign is constructed, the traffic sign is designed using software. The softcopy of the traffic design is termed as the template. The objective of this paper aims at extracting traffic sign features from the templates to group them in their corresponding class even though there is intra-class variations. Intra-class variations of the traffic sign exists due to allowable differences mentioned in the previous paragraph. To our knowledge, this is the first paper addressing the intra-class variation challenge in detail for the area of traffic sign symbol recognition. We would like to emphasize that the symbols used in this paper are computer generated images for traffic sign templates. From this point onwards,

\* Author for correspondence

we shall use the term classification in the context of traffic sign symbol recognition.

We look at traffic sign feature extraction with the aim of minimizing intra-class variability while being able to distinguish traffic signs from different classes at the same time. The features are attributes that will be fed into the classifier’s model. Examples of variability within class are depicted in Figure 1. A normal adult be able to recognize the sign as ‘bend-left’ regardless of the origin country of the sign. In contrast, in order for a computer to recognize the sign, the computer may require massive number of training samples. This is due to intra-class variability such as rim shape, background color, rim thickness, the shape of the arrow’s tail and head. A training process can be exhaustive if the number of training data is too big or the features selected are unsuitable. We employed knowledge of prominent features of the traffic signs to extract low number of features for high classification accuracy.



**Figure 1.** (a) Examples of ‘bend left’ sign from 22 countries. (b) Highlights of some intra-class spatial differences.

Section 2 of this paper reviews the research on visual based traffic sign recognition; Section 3 details the feature proposed and their extraction technique; Their application in classification is mentioned in Section 4; Experimental results of classification using various classifiers are presented in Section 5; Lastly, this paper is

concluded in Section 6.

## 2. Literature Review on Traffic Signs Recognition

Past and recent developed for automatic traffic sign recognition concentrated on resolving environmental variations such as weather, occlusion, resolution or/ and shape transformation. Testing were performed on real-traffic signs from specific countries, either self-captured<sup>3-10</sup>, using publicly available database<sup>11-13</sup> or extracted from the internet<sup>14</sup>. There is also an emerging trend to test a developed algorithm using real-world traffic sign images captured from different countries<sup>15-19</sup>. However, the researches on traffic sign recognition to-date have not specifically addressed the intra-class challenge.

A fully automated traffic sign recognition system usually takes a three-step approach, which is traffic sign detection; traffic sign feature extraction and traffic sign classification. The first step in the process is traffic sign detection for segmentation<sup>20</sup>. Input image that represents real-world is grabbed from the camera. Color is mostly used in segmentation as opposed to edges as a road background tends to create more ‘noisy’ edges. That noise appearance is accentuated if a traffic sign is a far.

The content within a traffic sign is accentuated by the shape of the object itself. Normally, the segmented traffic signs are converted to binary images<sup>21</sup> or grayscale images<sup>22</sup>. Meaning of a sign is given in form of symbol, number or alphabet. Thus, color is insufficient for recognition. Some traffic signs do not contain any object but the meaning of the traffic sign is given by the shape of its rim only, such as the ‘give way’ sign. Background color of the traffic sign is usually white or yellow to contrast the traffic sign object from the traffic sign itself. Thus, shape is an essential property for traffic sign recognition. Shape information can be gathered either from the edges<sup>3</sup>.

The next factor that affects the traffic sign recognition result is the database used for experiment. There are some publicly available databases, specific to a country. They consist of images taken from countries such as Germany, Sweden, Belgium, Netherlands, France and United States<sup>23</sup>. There are also databases originating from Asian countries such as Japan<sup>4,24</sup>, Taiwan<sup>6,25</sup> and Korea<sup>26</sup>. Most of the databases address partial or all of the environmental variations such as perspective distortions,

lighting, weather, occlusion and degradation. The most recent trend is the utilization of the German Traffic Sign Recognition Benchmark (GTRRB)<sup>27-28</sup>. Since the training and test images originates from a single country, intra-class challenge is not addressed.

Another group paced one step ahead by employing databases from two countries for training and used traffic sign from another country for testing<sup>17</sup>. Traffic signs from Spain and Czech Republic were used for training process while traffic signs from Britain were used for testing. Such experimental setup is uncommon and should be adopted more widely. In their work, histogram of pictogram was used as the feature vector. Their classification using Self-Organizing Map (SOM) yielded close to 100% for non-rotated signs. In their case, the structural challenge that increases intra-class challenge is not explicitly mentioned.

There are two main approaches in feature extraction, namely global and local features. In global feature extraction, constant size of the training images<sup>14-15</sup> is required. For instance, Histogram-of-Gradient (HOG) had been used and the dimension of the feature correlates with the size of the input image<sup>15-16, 24</sup>. A combination of HOG with Distance Transform (DT) and feature selectors' techniques with the Random Forest classifier yielded highest classification accuracy at 97% using the GTSRB database<sup>12</sup>. In<sup>29</sup> employed the Principal Component Analysis (PCA) and the Euclidean distance measurement. Their technique yielded 97.9% for speed limit signs. When the traffic sign is rotated, the result dropped to 80%. Neural networks such as the Convolution Neural Network<sup>13,28</sup>, Multilayer Perceptron<sup>30</sup> and Adaptive Resonance Theory (ART1)<sup>21</sup>, had been evaluated in the traffic sign recognition challenge. Another feature employed is the Fourier Transforms (FT)<sup>22</sup> to form a periodic representation of a contour. The sum-of-square differences were used for classification. The accuracy is 95.4% and it distinguishes seven classes of traffic signs. Variants of the SVM had also been employed in traffic sign recognition<sup>10-12,31</sup>. From our observation, the training data plays an important role in determining the algorithm robustness.

In local feature extraction, localized geometrical attributes are useful to identify prominent features of the traffic sign. It is not widely employed as a knowledge base of the object of interest has to be setup. An example of such approach had been employed for recognition of blue traffic sign containing different sets of arrows<sup>4</sup>. A knowledge

base was setup for the arrow signage using the decision tree. The accuracy rate was 81% for 13 classes containing various arrow patterns. If number of symbol that has to be recognized is greater, then the tree would increase in its complexity. Authors in<sup>3</sup> addressed segmentation issue from geometrical aspect by proposing optimal enclosure for pre-processing. Classification was done using template matching. However, template matching might not be flexible when it comes to addressing intra-class variability. Authors in<sup>31</sup> portrayed that certain arrow looking objects has similar geometrical attributes. Due to poor resolution of test inputs, they proposed to categorize them under the same class instead of differentiating them.

We wish to highlight the ability and simplicity of employing global and local features to solve intra-class variations. Although our experiments were based on traffic sign templates, we hope that it serves as a stepping stone towards assimilation of techniques in overcoming intra-class and environmental challenges for real-world road images.

### 3. Spatial Features to Normalize Intra-Class Variations of Traffic Signs

If an  $N$  number of features can be extracted from a raw image, the probability of success would depend on percentage of good features over bad features used. We argue if there are more good features used than poor features, then it would improve the probability of good feature selection  $P(x)$ , as explained in Equation (1). Better features would improve the recognition rate and reduce the number of features required. Our work concentrates on good feature extraction for symbols in the traffic sign templates. We hereby termed the symbol as object. The first step is traffic sign object segmentation and then its feature extractions.

$$P(x) = \frac{\binom{k}{x} \binom{N-k}{n-x}}{\binom{N}{n}} \quad (1)$$

$N$  = the total number of features.

$n$  = the number of features employed.

$k$  = the number of good features.

$x$  = the number of good features employed.

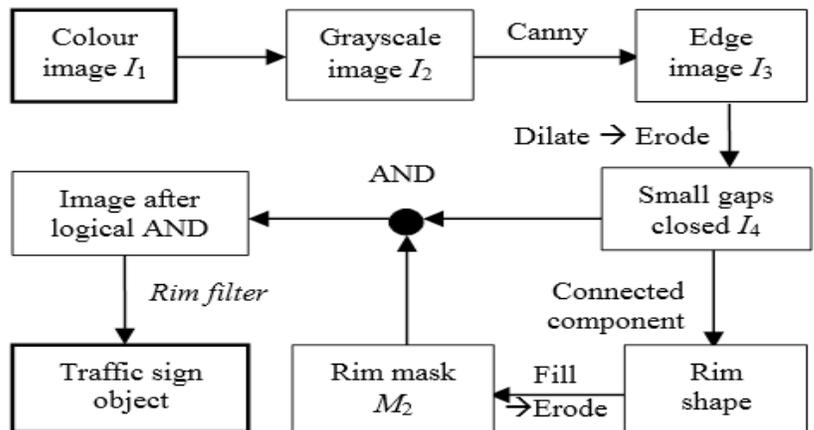


Figure 2. Traffic sign object segmentation.

### 3.1 Traffic Sign Object Segmentation and Rim Shape Recognition

The rim of a traffic sign with respect to its category is generalized as in Table 1. The scope of this paper is to classify traffic sign templates that fall under the warning category and to identify the category for other shapes. In short, it has to recognize the following rim shapes: Upright triangle, rhombus, downright triangle, octagon and circle.

Table 1. Traffic sign shapes and their category

Shape	Category	Examples
Upright triangle	Warning	
Rhombus	Warning	
Downright triangle	Give way	
Circle	Prohibitive/mandatory	
Octagon	Stop	

Firstly, the color image  $I_1$  was converted from RGB to grayscale image  $I_2$ . Then, the binary image  $I_3$  was extracted using the Canny edge detection. Small gaps were closed by dilating and then erosion using identical structure element as mask  $M_1$ . Connected component labelling on the binary image was performed ( $I_4$ ). The extreme outer contour of a traffic sign template is the rim shape. That contour is filled and eroded to act as a mask  $M_2$  for object shape extraction. After a logical AND operation of

$M_2$  with the earlier binary image  $I_4$ , the remaining pixels should belong to the traffic sign object only. The process is summarized in Figure 2.

We highlighting a rim filter to improve the segmentation.

Rim filter:

- An inspected contour has similar aspect ratio as the extreme outer contour.
- An inspected contour shares similar mid  $x-y$  coordinate as the extreme outer contour.
- An inspected contour must be relatively big compared the extreme outer contour.
- If the contour fulfill the three rules above, it is compared against the extreme outer contour mask using binary template matching. It is discounted as the object of interest if their shape matches.

Occasionally, there is some rim shapes preserved after logical AND. The remaining pixels after passing through the rim filter should exclusively belong to the final traffic sign object. The same binary template matching used in the rim filter was also used to distinguish the shape of

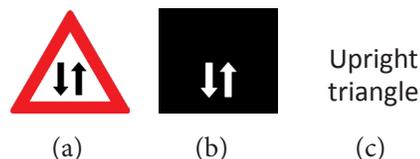


Figure 3. (a) Original image. (b) Traffic sign object. (c) Rim shape class.

### 3.2 Traffic Sign Feature Extraction

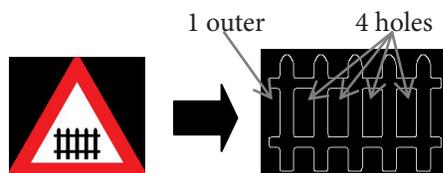
There are eleven main features extracted to differentiate inter-class objects while normalizing specific features of

intra-class objects. The features used are summarized in Table 2. Due to sub-features, there are a total of 22 features ( $f_{o1}$  to  $f_{o22}$ ). The features are fusion of global and local information. Global features gives inter-class discrimination while local features minimizes intra-class differences. The earlier rim shape class is  $f_{o1}$ . We extract the subsequent features in the order shown in Table 2. Each traffic sign image used for training and testing would be assigned a numerical value for each feature. The value is either extracted from the image itself or given a constant value if a feature is irrelevant to the traffic sign object.

**Table 2.** Global and local features for classification

No.	Feature	Global/ local
1	Traffic sign rim shape, $f_{o1}$	global
2	Number of contours, $f_{o2}$	global
3	Number of holes, $f_{o3}$	global
4	Horizontal similarity, $f_{o4}$	global
5	Vertical similarity, $f_{o5}$	global
6	Height to width ratio, $f_{o6}$	global
7	x-y direction of the mass center, $f_{o7} - f_{o8}$	global
8	Number of concave points, $f_{o9}$	global
9	Existence of strong vertical convexity defect, $f_{o10}$	local
10	Number of concavity by quadrant, $f_{o11} - f_{o14}$	local
11	Number of concavity by compass direction, $f_{o15} - f_{o22}$	local

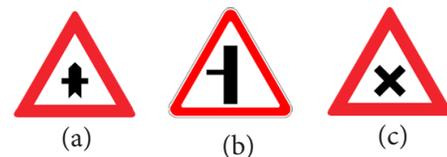
A fresh set of connected component labeling was performed after traffic sign object segmentation. The number of contours  $f_{o2}$  is counted based on the numbers of extreme outer contour the segmented image. The number of holes  $f_{o3}$  refers to the holes inside all extreme outer contours. The number of holes is obtained by subtracting the numbers of extreme outer contour from the total number of contours. Figure 4 shows the outer contour and the holes for a traffic sign object.



**Figure 4.** Outer contour and holes for a traffic sign object.

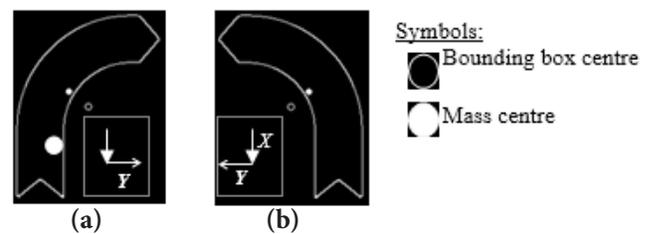
The next features are the traffic sign's horizontal  $f_{o4}$  and vertical  $f_{o5}$  similarity index. Certain traffic sign would have

same shape after folding in half horizontally, vertically or in both directions. After being folded, the first and the second half segment are compared using binary template matching, as mentioned in Section 3.1. Figure 5 shows examples of traffic signs object that have shapes which are similar when they are folded horizontally, vertically or in both directions.



**Figure 5.** The object's shape is similar when it is folded. (a) Vertically. (b) Horizontally. (c) Both directions.

The bounding box for the segmented object is used to compute the aspect ratio for the feature  $f_{o6}$ . The next feature is the x-y direction of the mass center that is measured against the center of the bounding box. The bounding box is the smallest upright rectangle that encompasses a traffic sign object. This feature is applicable to objects containing one outer contour only. It is an important feature to distinguish objects with similar shape but with different meaning as shown in Figure 6. The mass center is computed using central moments, where  $(\bar{x}, \bar{y})$  is the coordinate of the mass center and  $(x_b, y_b)$  is the center of the bounding box. The x-y directions were obtained by identifying the path taken by the mass center to the bounding box's center. Their quantifications are given in Equations (2)-(3).



**Figure 6.** Directions of mass center from bounding box center for. (a) The 'bend right' sign. (b) The 'bend left' sign.

$$f_{o7} = \begin{cases} 1 & ; \bar{x} > x_b \\ 0 & ; \bar{x} = x_b \\ -1 & ; \bar{x} < x_b \end{cases} \quad (2)$$

$$f_{o8} = \begin{cases} 1 & ; \bar{y} > y \\ 0 & ; \bar{y} = y \\ -1 & ; \bar{y} < y \end{cases} \quad (3)$$

### 3.3 Shape Convexity for Local Feature Extraction

Majority of the traffic sign objects have some concavities. Exceptions are objects with prominent shapes such as circle, oval and rectangle. The convex hull of each outer contour and its respective concave points were used in extraction of the subsequent features. Concavity is computed by finding the convexity defects of each contour. Convexity defect was chosen as the feature due to its robustness to fluctuation of image resolution. The depth of the concave points with respect to the convex hull were captured. The quantity, depth, location and direction of the concave points were used as features ( $f_{o9} . f_{o22}$ ). Extraction of the concavity related points are illustrated Figure 7. Let  $j$  be the number of convexity defect. The area of the convexity defect is used to calculate the depth  $d_j$ , as explained in Equations (4)-(5).

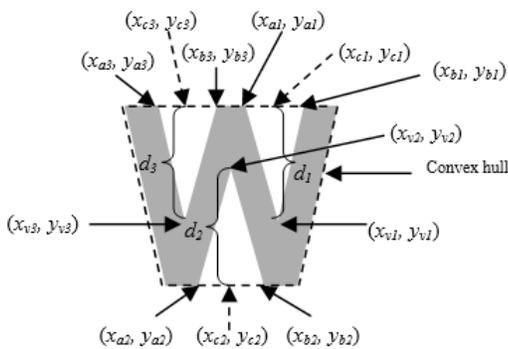


Figure 7. Three sets of points used in calculated depth of three convexity defects ( $j = 1, 2, 3$ ).

$$x_{cj} = \frac{x_{aj} + x_{bj}}{2} \quad (4)$$

$$y_{cj} = \frac{y_{aj} + y_{bj}}{2}$$

$$d_j = \sqrt{(x_{vj} - x_{cj})^2 + (y_{vj} - y_{cj})^2} \quad (5)$$

If a depth  $d_j$  is smaller than a threshold  $t_a$ , then it will be ignored. Subsequently, all concave point detected need to pass through a concavity filter to generalize irregularities due to intra-class variation, as shown in Figure 8.

#### Concavity Filter:

- There is only one extreme outer contour  $f_{o2}$  for the segmented object and.
- There are more than one concave point  $f_{o9}$  and.
- If the aspect ratio  $f_{o6}$  of the bounding box is greater than a threshold  $t_b$ :
  - $y_c$  is near to the bottom of the bounding box.
  - The concave point is ignore if  $d_j$  is very small compared with the deepest concave point  $d_{max}$  defined by threshold  $t_c$ .
- Else if the aspect ratio  $f_{o6}$  of the bounding box is smaller than a threshold  $t_b$ :
  - A concave point  $m$  has depth greater than a specific threshold,  $t_d$  and its  $x_c$  is located near the mid- $x$  position of the bounding box.
  - If there is concavity at the closed to the right or left side of the object, the side concave point is ignored if the depth is smaller than  $d_m$ .

Next, the traffic sign object is compartmentalized into four quadrants. The number of concave occurrence in each image quadrants is logged. This would result in four

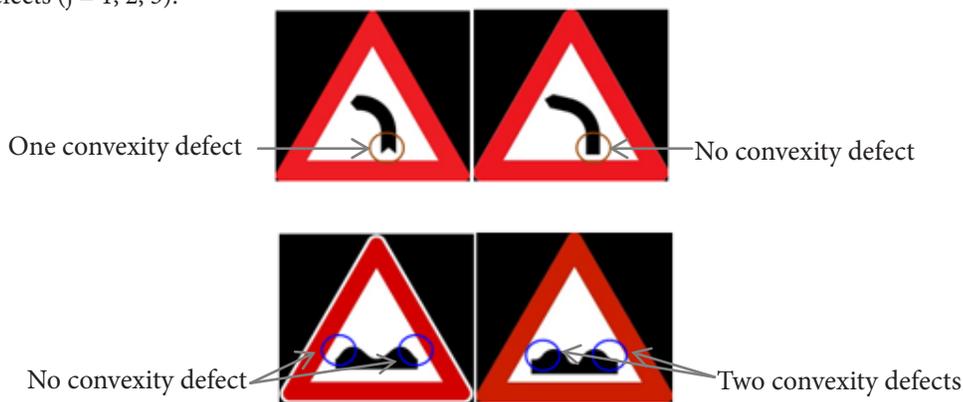
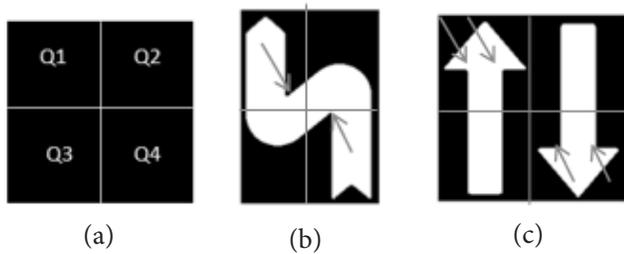


Figure 8. Intra-class irregularities for convexity defect.

sub-features  $f_{o11} - f_{o14}$  to indicate the number of concave point per quadrant, as describe in Figure 9.



**Figure 9.** (a) Division into four quadrants based on bounding box, Number of concave points for each quadrant. (b) Q1 = 1, Q2 = 0, Q3 = 0, Q4 = 1. (c) Q1 = 2, Q2 = 0, Q3 = 0, Q4 = 2.

Later, each concave direction is computed. The direction is the path from  $(x_{cp}, y_{cp})$  to the concave point  $(x_{vp}, y_{vp})$ . Eight regions are identified for each reference point. The reference point is  $(x_{cp}, y_{cp})$ . The angle of a concave point  $\theta$  and its  $x$ - $y$  directions were used to compute the directions of a concave point, as shown in Equations (6)-(7). The concave direction is determined based on the properties stated in Table 3. A buffer of  $10^\circ$  angle is allowed for north, south, east and west directions. A concave point is categorized into one concave direction. Each concave point is categorized to one of the eight sub-features  $f_{o15} - f_{o22}$ , which is a concave point's direction. Examples are illustrated in Figure 10.

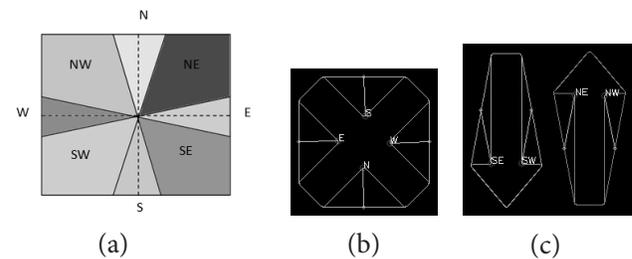
$$\theta = \tan^{-1} \left( \frac{y_{ci} - y_{vi}}{x_{vi} - x_{ci}} \right) \times \frac{180}{\pi} \tag{6}$$

$$y_{dir} = \begin{cases} 1 & ; y_{vi} \geq y_{ci} \\ -1 & ; y_{vi} < y_{ci} \end{cases} \tag{7}$$

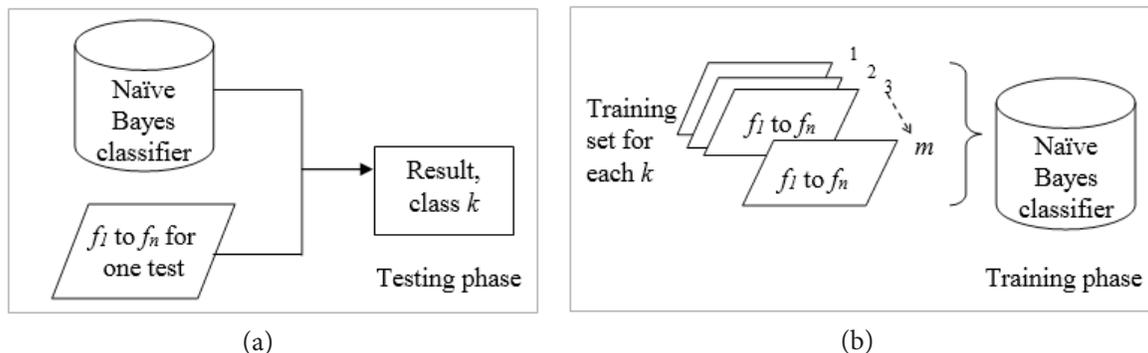
$$x_{dir} = \begin{cases} 1 & ; x_{vi} \geq x_{ci} \\ -1 & ; x_{vi} < x_{ci} \end{cases}$$

**Table 3.** Direction of concave point

$y_{dir}$	$x_{dir}$	$\theta$	Direction
any	1		East (E)
1	1	$\theta \leq 5^\circ$ $5^\circ < \theta < 85^\circ$	North-East (NE)
1	any	$\theta \geq 85^\circ$	North (N)
1	-1	$5^\circ < \theta < 85^\circ$	North-West (NW)
any	-1	$\theta \leq 5^\circ$	West (W)
-1	-1	$5^\circ < \theta < 85^\circ$	South-West (SW)
-1	any	$\theta \geq 85^\circ$	South (S)
-1	1	$5^\circ < \theta < 85^\circ$	South-East (SE)



**Figure 10.** (a) Illustration of the eight directional regions. (b) E = 1, S = 1, W = 1, N = 1. (c) SE = 1, SW = 1, NE = 1, NW = 1.



**Figure 11.** (a) Training inputs to build the Naïve Bayes classifier model. (b) Testing phase for each traffic sign object.

## 4. Classification Using the Spatial Features

Naïve Bayes is a probabilistic classifier using the known priory of a set of conditions that result in an event. The conditions are assumed to be independent of one another. An event can be predictable correctly only if most of the past conditions that define the event is accurate. Computation cost of the Naïve Bayes model can be cheaper in comparison to other machine learning techniques, if suitable features are used to build the classifier model. Its application had been showcased in<sup>8</sup>, where the authors used small raw image of fixed resolution (20 x 24) for classification. In their case, each pixel in the training data is considered as a feature element, which results in 480 attributes for each image. In contrast, we reduced the number of attributes to 22 units only. On top of that, we included intra-class variation and used low number of training images in our experiments.

Traffic sign objects contain prominent shapes that can use to build the knowledge base. We explain the application of feature  $f_{o1} - f_{o22}$  using the Naïve Bayes classifier here. In Figure 11,  $n$  denotes the number of features while  $m$  denotes the number of training sets to build the classification model.  $k$  refer to different classes of traffic sign object. The model built is used in classification of the

For each class  $k$ , there  $m$  sets of are  $f_n$  independent features. The training model is used in the testing phase where the probability obtained for each class is computed. The class with the highest probability is the winner, as described in Equation (8). Details about the Naïve Bayes classifier can be found here<sup>33</sup>.

$$\hat{k} = \arg \max_k P(k) \prod_{i=1}^n P(f_i | k) \tag{8}$$

The applications of the features  $f_n$  requires quantification of each feature. However, there are certain features that could not be quantified due to the original shape of the traffic sign object. For instance, number of concavity by quadrant  $f_{o11} - f_{14}$  is irrelevant for a circle because there is no concave point. In such cases, a constant will be assigned on the feature of such inputs. The constant value is given intuitively, in such a way that it emphasizes other features of the object.

## 5. Results and Discussions

### 5.1 Database and the Experimental Platform Setups

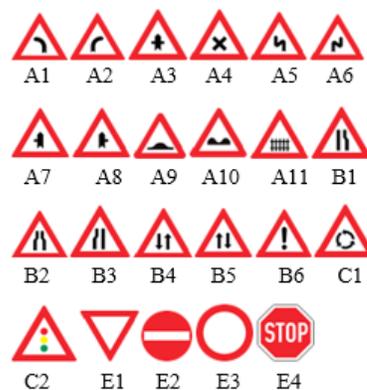
To date, there is no public database that systematically exhibits intra-class variability. For this study, we have

gathered traffic sign templates from the internet. The templates consist of traffic signs from various countries in order to capture intra-class variability of object shapes. The images are classified into twenty-three traffic sign classes ( $k = 23$ ). The categories and their corresponding classes are summarized in Table 4.

**Table 4.** Categories and classes of the database

Category	A	B	C	E
Class	1 to 11	1 to 6	1 to 2	1 to 4

The traffic sign templates originated from various European countries: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Netherlands, Norway, Poland, Russia, Slovakia, Spain, Sweden, Switzerland, Ukraine, and the United Kingdom. Category A, B and C are traffic signs under the warning category. Majority of their rim shapes is upright triangle. Traffic signs with rhombus shaped rim are also placed under the warning category. Traffic signs in Category A contains object with one extreme outer contour only. Traffic signs in Category B contain mainly object with two extreme outer contours while traffic signs under Category C contain object with three outer contours. Traffic signs in Category E are non-warning signs with rim shapes other than upward pointing triangle or rhombus. The classes from each category are concatenated to form the formal class for the traffic sign (Figure 12). The resolution of the original image ranges from 47 x 47 to 800 x 698. The average size of the images in the database is 478 x 432. Smaller images were enlarged during pre-processing phase. There are a total of 472 images in the dataset. The complete database can be obtained at <http://pesona.mmu.edu.my/~hlwong/TS.html>



**Figure 12.** Twenty-three formal classes in the dataset.

All experiments were conducted on a 64-bit notebook

with Intel(R) Core(TM) i5-2410M CPU@2.30GHz at 4GB RAM, running on Windows 7. The feature extraction procedure was written using Visual C++ 2010 along with OpenCV version 2.4.9. The WEKA version 3.7.11 data mining software was used for classification.

### 5.2 Classification using the Naïve Bayes Classifier

Firstly, the features computed were concatenated into a single ARFF file. 10-fold cross validation option in WEKA was used to evaluate the performance of the features proposed using the Naïve Bayes classifier. The data was divided into ten parts. Each part is hold in turn for testing. This means each part is used once for testing and used nine times for training. The same image would not be used for testing and training in the same iteration. Later, the average result is returned. The result is given in Figure 13 in the form of a confusion matrix. There are only three test inputs assigned to incorrect class, resulting in an accuracy of 99.4%.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- Classified as	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = A1
0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = A2
0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = A3
0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = A4
0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = A5
0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = A6
0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g = A7
0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h = A8
0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	i = A9
0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	j = A10
0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	k = A11
0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	l = B1
0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	m = B2
0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	n = B3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	o = B4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	p = B5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	2	0	0	0	0	0	q = B6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	r = C1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	s = C2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	t = E1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	u = E2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	v = E3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	w = E4

Figure 13. Confusion matrix for classification result using the Naïve Bayes classifier.

Among the three failure, two input failed due to poor resolution. One as extreme contours which are falsely connected visually. Another one has false concavities that appeared after image enlargement. The last input failed due to the uniqueness of its object shape which differs it from the nominal ‘other danger’ signs. Its original shape has only one outer contour while majority of the training sets for class B6 have two outer contours.

In the next experiment, we measured the performance of the Naïve Bayes classifier when the number of training data is the constraint. WEKA randomly divides the data into different proportions of training and test sets. The

result for traffic sign object classification is shown in Figure 14. The x-axis of the graph indicates the percentage of training data. If the value is 10, this means that 10% of the images in the database were used for training and remaining 90% were used for testing. This splitting is termed as 10:90 ratio. As shown in the graph, for the training-test input with 20:80 ratio upwards, the result improved drastically. This indicates that even with a small number of training set  $m$ , classification is high at 95%. Moving forward in the graph, no error was observed at 50:50 ratio. An image  $z$  that exhibits greater intra-class variability was not selected for testing by the random training-test input splitter. Its variation was normalized by other training inputs. From 60:40 ratio upwards, the number of test input classified wrongly is one ( $z$ ). There is deterioration in classification accuracy because the number of test input decreases while the number of incorrect classification output remains at one unit.

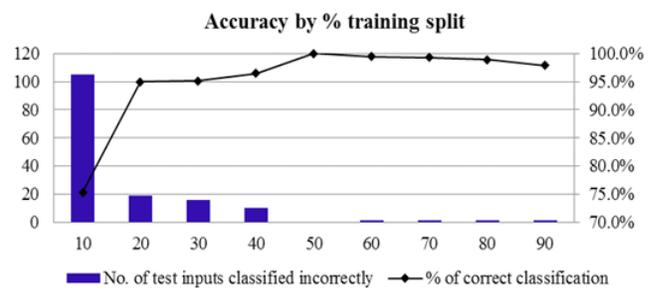


Figure 14. Naive Bayes classification accuracy for different proportions of training-test input quantity.

The average test-time per test input is 0.43 ms. In contrast, there is not much difference in the model building time. The model building time stayed low at 0.02 s even when the number of training sets fluctuates. Model building time is not critical as the model building is usually performed once only. If there is no update in the knowledge base of the model, then the model re-training is not required.

### 5.3 Performance Comparison using Other Classifiers

The features proposed were also tested using various classifiers. Thirteen classifiers, including the Naïve Bayes classifier, were used. Default WEKA software parameter settings were used in all of the classifiers tested except for AdaBoost and Random Committee, where each of them was combined with the Random Forest. Their accuracies

are summarized in Table 5. There is no big variation in test-time. The model building time is given in Table 6. 10-fold cross validation method was employed in these measurements. The Naïve Bayes classifier showed the best accuracy at 99.4%. Random Tree resulted in the lowest accuracy at 93%. The model building times for the K-Nearest Neighbors and the Random Tree were the best. This is followed by the Naïve Bayes at 0.02 s. However, the accuracies of the K-Nearest Neighbors and the Random Tree were much lower than the Naïve Bayes classifier, which are 93.2% and 93% respectively. The Multilayer Perceptron gave an accuracy of 97.5%, but the model building time is the longest, which is 20.86 s. The result shows that the Naïve Bayes classifier is the best classifier using the features proposed. It is computationally cheap and the accuracy yields the highest.

**Table 5.** Classification accuracies

Classifiers	Accuracy (%)
Naïve Bayes	99.4%
RandomCommittee+Random Forest	98.5%
AdaBoost+Random Forest	98.1%
BayesNet	97.9%
Naïve Bayes Tree	97.7%
LibSVM	97.5%
Multilayer Perceptron	97.5%
RandomForest	97.2%
Best-First Tree	95.8%
Radial Basis Function Networks	95.6%
Decision table- Naïve Bayes Hybrid	95.3%
K-Nearest Neighbours	93.2%
RandomTree	93.0%

**Table 6.** Model building time using various classifiers

Classifiers	Time (s)
K-nearest neighbours	0
RandomTree	0
Naïve Bayes	0.02
BayesNet	0.02
RandomForest	0.03
AdaBoost+Random Forest	0.08
Best-First Tree	0.31
RandomCommittee+Random Forest	0.36
Radial Basis Function Networks	0.39
Naïve Bayes Tree	4.23
LibSVM	5.99
Decision table- Naïve Bayes Hybrid	11.9
Multilayer Perceptron	20.86

## 6. Conclusions

This work is intended for visual-based classification of traffic sign templates at the presence of intra-class variations. The classification outcome can be used for an automated semantic annotation of the traffic sign templates by a computer. We proposed shape-based features to enhance variability of inter-class traffic sign objects while reducing the variability of intra-class traffic sign objects. They consist of global and local features extracted from the traffic sign templates. The application of the features was depicted through the implementation using the Naïve Bayes classifier. The best classification accuracy of 99.4% at an average test-time of 0.43 ms per test input was achieved using the Naïve Bayes classifier. We also demonstrated that 95% accuracy can be achieved even though only 20% of the data were used for training. There are visible intra-class shape variations among the feature sets for class  $k$ . The result indicates that the features proposed are versatile to intra-class variations. We also highlight that the number of feature  $f_n$  and the number of training set  $m$  used to achieve the positive results are low.

With this framework, we hope to move into the area of real-world traffic sign recognition, focusing on good features to extract. In real-world, the traffic sign would be affected by various challenges such as resolution, motion blur, lighting and deformation. Thus, segmentation is the next great challenge. Our aim is to minimize the number of training image and to reduce the complexity of training process through identification of good features.

## 7. Acknowledgements

The authors would like to thank University of Malaya for supporting this work under the PPP grant: PV075-2011A. We thank Google and various sources from the internet from which the experiment dataset was built and had been used for research purposes only. We also wish to express our appreciation to OpenCV and University of Waikato (WEKA) for being able to use their software tools freely.

## 8. References

1. Convention on road signs and signals. Vienna, Economic Commission for Europe. Inland Transport Committee; 1968.

2. Arahan Teknik (Jalan) 2A/85 – Standard traffic signs. Malaysia: Ministry of Works Malaysia; 1985.
3. Hu S. Robust traffic sign shape recognition using geometric matching. *IET Intell Transp Syst.* 2009; 3(1):10–8.
4. Alsibai MH, Hirai Y. Real-time recognition of blue traffic signs designating directions. *Int J ITS Res.* 2010; 8:96–105.
5. Escalera S, Pujol O, Raveda P. Traffic sign recognition system with  $\beta$ -correction. *Mach Vision Appl.* 2010; 21:99–111.
6. Hsu S-H, Huang C-L. Road sign detection and recognition using matching pursuit method. *Image Vision Comput.* 2001; 19:119–29.
7. Piccioli G, Micheli ED, Parodi P, Campani M. Robust method for road sign detection and recognition. *Image Vision Comput.* 1996; 14:209–23.
8. Muhammad AS, Lavesson N, Davidsson P, Nilsson M. Analysis of speed sign classification algorithms using shape based segmentation of binary images. *Lect Notes Comput Sc.* 2009; 5702:1220–7.
9. Gao XW, Podladchikova L, Shaposhnikov D, Hong K, Shevtsova N. Recognition of traffic signs based on their colour and shape features extracted using human vision model. *J Vis Commun Image Represent.* 2006; 17:675–85.
10. Timofte R, Zimmermann K, Gool LV. Multi-view traffic sign detection, recognition, and 3D localisation. *Mach Vision Appl.* 2014; 25(3):633–47.
11. Yaun X, Hao X, Chen H, Wei H. Robust traffic sign recognition based on colour global and local oriented edge magnitude patterns. *IEEE Trans Intell Transp Syst.* 2014; 15(4):1466–77.
12. Zaklouta F, Stanculescu B. Real-time traffic sign recognition using tree classifiers. *IEEE Trans Intell Transp Syst.* 2012; 13(4):1507–14.
13. Jin J, Fu K, Zhang C. Traffic sign recognition with hinge loss trained convolutional neural network. *IEEE Trans Intell Transp Syst.* 2014; 15(5):1999–2000.
14. Gonzalez, Bergasa LM, Yebes JJ. Text detection and recognition on traffic panels from street-level imagery using visual appearance. *IEEE Trans Intell Transp Syst.* 2014; 15(1):228–38.
15. Lu K, Ding Z, Ge S. Sparse representation-based graph embedding for traffic sign recognition. *IEEE Trans Intell Transp Syst.* 2012; 13(4):1515–24.
16. Greenhalgh J, Mirmehdi M. Real-time detection and recognition of road traffic signs. *IEEE Trans Intell Transp Syst.* 2012; 13(4):1498–506.
17. Prieto MS, Allen AR. Using self-organising map in the detection and recognition of road signs. *Image Vision Comput.* 2009; 27:673–83.
18. Ruta A, Li Y, Liu X. Real-time traffic sign recognition from video class-specific discriminative features. *Pattern Recogn.* 2010; 43:416–30.
19. Ruta A, Li Y. Robust class similarity measure for traffic sign recognition. *IEEE Trans Intell Transp Syst.* 2010; 11(4):846–55.
20. Gomez-Moreno H, Maldonado-Bascon S, Gil-Jimenez P, Lafuente-Arroyo S. Goal evaluation of segmented algorithms for traffic sign recognition. *IEEE Trans Intell Transp Syst.* 2010; 11(4):917–30.
21. Escalera ADL, Armingol JM, Mata M. Traffic sign recognition and analysis for intelligent vehicles. *Image Vision Comput.* 2003; 21:247–58.
22. Larson F, Felsberg M, Forssen PE. Correlating fourier descriptor of local patches for road sign recognition. *IET Comput Vis.* 2011; 5(4):244–54.
23. Mogelmose A, Trivedi MM, Moeslund TB. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspective and survey. *IEEE Trans Intell Transp Syst.* 2012; 13(4):1484–97.
24. Ruta A, Porikli F, Watanabe S, Li Y. In-vehicle camera traffic sign detection and recognition. *Mach Vision Appl.* 2011; 22:359–75.
25. Fang CY, Fuh CS, Yen PS, Cherng S, Chen SW. An automatic road sign recognition system based on a computational model of human recognition processing. *Comput Vis Image Und.* 2004; 96:237–68.
26. Vavilin A, Jo K-H. Graph-based approach for robust road guidance sign recognition from differently exposed images. *J Univers Comput Sci.* 2009; 15(4):786–804.
27. Stallkamp J, Schlipsing M, Salmen J, Igel C. The German traffic sign recognition benchmark: A multi-class classification competition. *International Joint Conference Neural Network;* 2011. p. 1453–60.
28. Stallkamp J, Schlipsing M, Salmen J, Igel C. Man vs computer: Benchmarking machine learning algorithm for traffic sign recognition. *Neural Networks.* 2012; 32:323–32.
29. Fleyeh H, Davami E. Eigen-based traffic sign recognition. *IET Intell Transp Syst.* 2011; 5(3):190–6.
30. Nguwi Y-Y, Kouzani AZ. Detection and classification of road signs in natural environments. *Neural Comput and Applic.* 2008; 17:265–89.
31. Maldonado-Bascon S, Lafuente-Arroyo S, Gil-Jimenez P, Gomez-Moreno H, Lopez-Ferreras F. Road sign detection and recognition based on support vector machines. *IEEE Trans Intell Transp Syst.* 2007; 8(2):264–78.
32. Liu Y-S, Duh D-J, Chen S-Y, Liu R-S, Hsieh J-W. Scale and skew-invariant road sign recognition. *Int J Imag Syst Tech.* 2007; 17(1):28–39.
33. Timofte R, Tuytelaars T, Gool LV. Naive Bayes image classification: Beyond nearest neighbors. *Lect Notes Comput Sc.* 2012; 7724:689–703.