VIETNAM NATIONAL UNIVERSITY OF HCMC

INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

EXTEND TRAFFIC SIGNS DETECTION AND RECOGNITION ALGORITHM IN NIGHTTIME IN VIET NAM

By
Vu Tuan Anh

A thesis submitted to the School of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Science

Ho Chi Minh City, Vietnam
June, 2016

EXTEND TRAFFIC SIGNS DETECTION AND RECOGNITION ALGORITHM IN NIGHTTIME IN VIET NAM

APPROVED BY:

__________________________________,

__________________________________,

____

____

____

THESIS COMMITTEE

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to Dr. Ha Viet Uyen Synh, my supervisor, for advising me during the research process, as well as providing me essential knowledge about image processing. By his enthusiasm and immense knowledge, He also helped me to come up with a solution to my research problems and shared with my experiences in researching methods. Beside my supervisor, I want to express my appreciation to other members of the research group who helped me with during the research process and assisted me in capturing and collecting the video traffic datasets. Finally, I own a big thank to School of Computer Science and Engineering for making a good condition for me to accomplish my thesis.

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Nowadays, traffic sign recognition has several useful applications, ranging from simple Driver Assistance Systems (DAS) to full-fledged Autonomous vehicles. In DAS, the Detection and Recognition of Traffic Signs are critical; they can help reduce accidents by alerting drivers of the presence of Traffic indications that they may have missed due to adverse atmospheric conditions or by their bad positioning in the road, etc. However, many recent studies just show the Recognition of traffic signs in the best conditions, without external factors influence. However, the city in Viet Nam is rapidly developing, traffic density overcrowded with the exchange of good along both sides, the weather, the state of traffic signs, the light levels will affect to detect the traffic signs. Especially to detect the Traffic signs in low light condition is tough.

This thesis provides method of Traffic Signs Detection and Recognition in Viet Nam with three main techniques consist of color segmentation, shape recognition and Support Vector Machine (SVM).

Keywords: object detection, traffic sign recognition, contour, morphology, color segmentation, driver assistance systems, support vector machine.

CHAPTER 1
INTRODUCTION

1.1 Problem statement

Along with the development of the scientific achievements of modern technology, the dynamics of the market economy mechanism, the advanced of life, the traffic problems have also been improving and developing to strongly contribute to the development of our society.

From the actual traffic condition in Vietnam, the majority of accidents are caused by the drivers cannot control the speed, do not follow the traffic command, not observed or recognized the traffic signs and traffic signals. The main cause leading to the above problem is the driver fatigues, lack of concentration and is impacted by the other factors.

Moreover, the application of high technology in the creation of self-propelled vehicles is essential trends of social development. Therefore, the main topic of my thesis is “Extend Traffic Signs Detection and Recognition Algorithm in Nighttime in Viet Nam”.

1.2 Objectives and Scopes

The overall aim is to develop a system that can be used for Traffic Sign Detection and Recognition. This system can assist driver by automatically detecting and classifying one or more traffic signs from a complex scene.

This project will focus on the detection and recognition of three type traffic signs (dangerous signs, mandatory signs, prohibitory signs), viewed at the street level and from a frontal perspective.

The traffic signs can be loaded from a stored image or video, and can also be from a live feed retrieved from a connected camera.

1.3 Assumption and Solution

Because of time limitation, knowledge and real condition, this system also has some limitations such as:

- ✓ Not included the case of traffic signs in bad weather: fog, rain.
- ✓ Camera will always suffer from vibrations.
- ✓ Small database of traffic sign images.

1.4 Structure of thesis

This part will describe the structure of thesis:

Chapter 1 describes an introduction, point out problems and purpose of doing this thesis.

Chapter 2 reviews all necessary theories and knowledge to solve this problem and applications of this system in real life.

Chapter 3 presents some existing method for this system.

Chapter 4 presents proposed method and explain function of each phase.

Chapter 5 show results and discussion of my proposed method.

Chapter 6 describes conclusion and future work for my system.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview of the Traffic Signs

2.1.1 What are the Traffic signs?

Traffic signs or road signs are signs erected at the side of roads which define a visual language that can be interpreted by the drivers and road users. They represent the current traffic situation on the road, show the danger and difficulties around the drivers, give warnings to them, and help them with their navigation by providing useful information that makes the driving safe and convenient [1].

Road and traffic signs are designed to be easily recognized by drivers mainly because their shapes and colors are readily distinguishable from their surroundings [2].

Figure 1. An image of a traffic sign in Vietnam

2.1.2 Properties of road and traffic signs

Road signs or traffic signs are characterized by some features which make them recognizable and visible on the environments:

- ✓ Road signs are produced, designed, installed and constructed according to strict regulations.
- ✓ They are conceived in fixed 2-D shapes such as triangles, circles, octagons, or rectangles [3] [4].
- ✓ The colors of the signs are chosen to contrast with the surroundings, which make them easily recognizable by drivers [5].
- ✓ The information of each part on the sign has difference color.
- ✓ The color of the paint which consists of signs must correspond to a specific wavelength in the visible spectrum.
- ✓ They may contain a pictogram, a string of characters or both [6].
- ✓ They can appear in different conditions, including partly occluded, distorted, damaged and clustered in a group of more than one sign [3] [6].

2.1.3 Traffic signs in Viet Nam

Vietnamese traffic signs may be either ideogram-based which contain simple ideograph to express the meaning of the signs, or text-based where the content of the signs can be either text or arrows or other symbols. They always appear in the four basic shapes: rectangles, triangles, circles and octagons with five types of traffic sign such as dangerous signs, prohibitory signs, supplementary signs, mandatory signs, and indicatory signs.

Vietnamese traffic sign system has totally 48 indicatory signs, 9 mandatory signs, 39 prohibitory signs, and 46 dangerous signs.

❖ **Prohibitory signs:**

They are used to prohibit certain types of maneuvers or certain kinds of traffic. The no entry, no parking, and speed limit signs belong to this category. Normally, they are designed in a circular shape with a thick red rim, a white interior, and black pictograms or characters. There are few exceptions: the STOP sign is an octagon with a red background, white rim and white character, the NO PARKING and NO STANDING signs have a blue background instead of white.

Figure 2. Prohibitory signs

❖ **Mandatory signs:**

They are characterized by a complete blue circle and a white arrow or pictogram. They control the actions of drivers and road users. Signs ending obligation have a diagonal red slash.

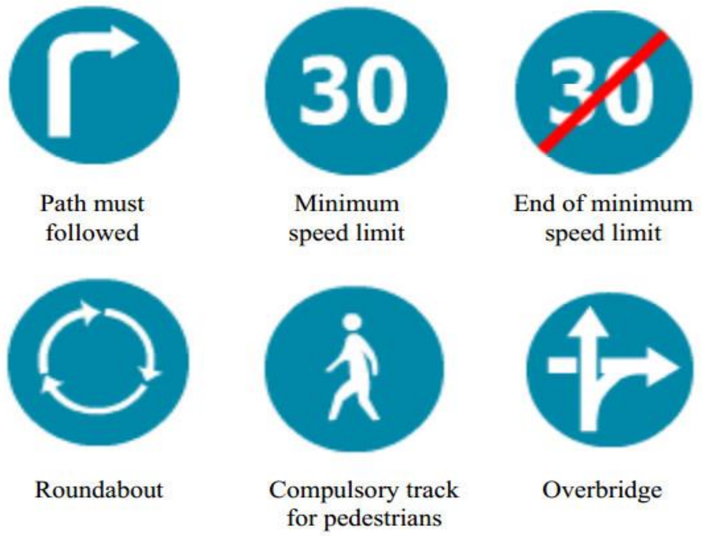

Figure 3. Mandatory signs

- **<u>Indicatory signs:</u>**

They are usually in rectangular or square shape with a blue background to alert road users to know the necessary orientation or useful things in the journey.

Figure 4. Indicatory signs

❖ **Dangerous signs:**

Traffic warning signs are a type of traffic sign which indicates a hazard ahead on the road that may not be readily apparent to a driver. These signs indicate a need for additional care by road users, and may require a decrease in speed or other maneuvers to ensure the safety of themselves, other road users, pedestrians or animals. They are characterized by an equilateral triangle with a thick red rim and a yellow interior. A pictogram, usually in black, is used to specify different warnings. Vietnamese traffic signs system has totally 246 signs belong to this type.

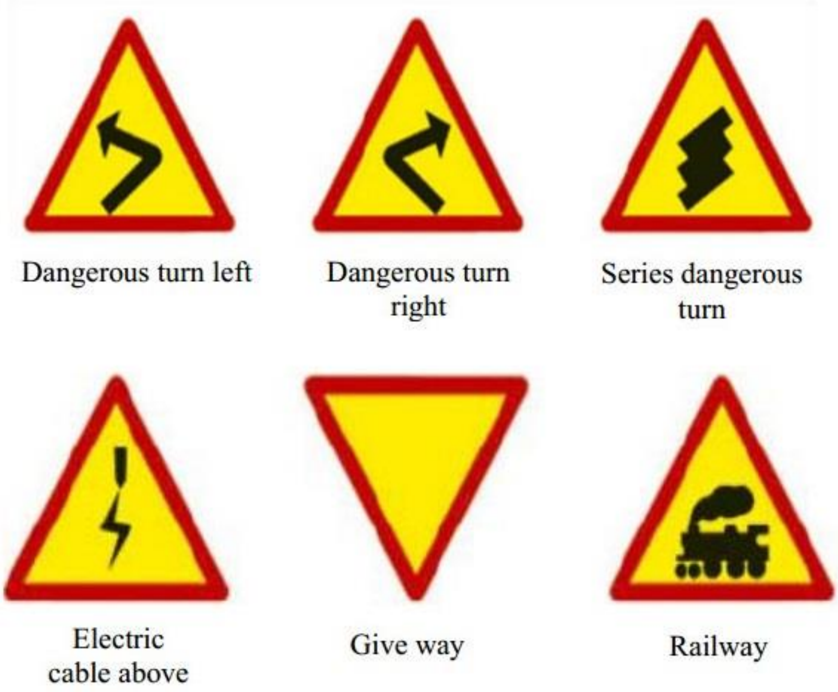

Figure 5. Dangerous signs

❖ **Supplementary signs;**

They are usually in rectangular or square shape, placed in conjunction with an indicatory sign, mandatory sign, prohibitory sign and dangerous sign for further explanations of these signs or can be used independently.

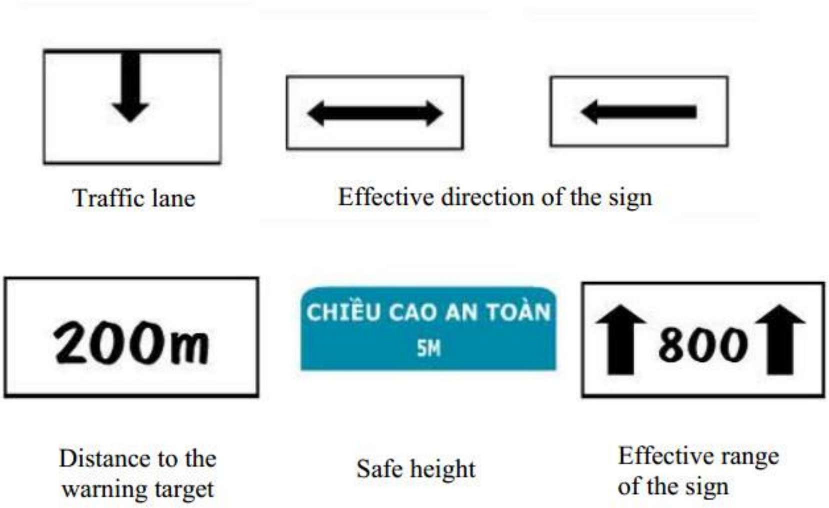

Figure 6. Supplementary signs

2.1.4 Potential difficulties

Because of the complex environment of the roads and the scenes around them, road signs can be found in different conditions, and therefore the detection and recognition of these signs may face one or more of the following difficulties:

The information of the sign fades with time as a result of the prolonged exposure to the sunlight, and the reaction of the paint with the pollutants in the air [7].

Figure 7. Faded signs

The visibility of traffic signs is affected by the weather conditions like fog, rain, clouds and snow, and other parameters like local light variations such as the direction of light, the strength of the light depending on time of the day and season, and shadows generated by other objects [6] [8]. The color information is very sensitive to the variations of the light conditions such as shadows, clouds, and the sun [7] [9].

It can be affected by the illuminant color (daylight), illumination geometry, and viewing geometry.

Figure 8. Inclement weather and lighting geometry

The presence of the obstacles in the scene, like trees, buildings, vehicles and pedestrians [9]. The presence of objects similar in color and/or shapes to the road signs in the scene under consideration, like buildings, or vehicles [7]. They could be similar to road sign by color, shape or even both of them. Signs may be found disoriented, damaged or congested. The size of the sign depends on the distance between the camera and the sign. The traffic signs may appear rotated due to the image orientation.

If the image is acquired from a moving car, then it is often suffering from motion blur and car vibration [10]. The absence of standard database for evaluation of the existent classification methods [11] [12].

Figure 9. A case of occluded sign and similar objects in the scene

2.2 Application of Detection and Recognition of the Traffic signs in Driver Assistance Systems (DAS)

Nowadays, the cities in Viet Nam is rapidly developing, traffic density overcrowded with the exchange of good along both sides, the weather, the state of traffic signs, the light levels will affect to detection of the traffic signs of the driver. The driver maybe is distracted, unfocused due to sleepy, drunk, surrounding atmosphere, etc. so they can cause accidents during driving. Therefore, the Driver Assistance Systems plays a critical role in preventing, restricting the occurrence of a traffic accident, save the driver and other people's live and enhance effectively driving. It provides necessary information to the driver about the traffic signs, the mandatory or prohibits commands that the drivers must follow, or the warning and direction are very difference.

Today, the system support the driver is smarter to ensure the driver's safe. The traffic signs detection and recognition system are becoming popular under the development of car manufacturing company in the world. This system consists of three modules: camera, CPU, and the warning system via screen and sound. In this system, the camera placed at the top of the car will capture the traffic signs, then the program analyzes and recognizes the image and show the result on the screen.

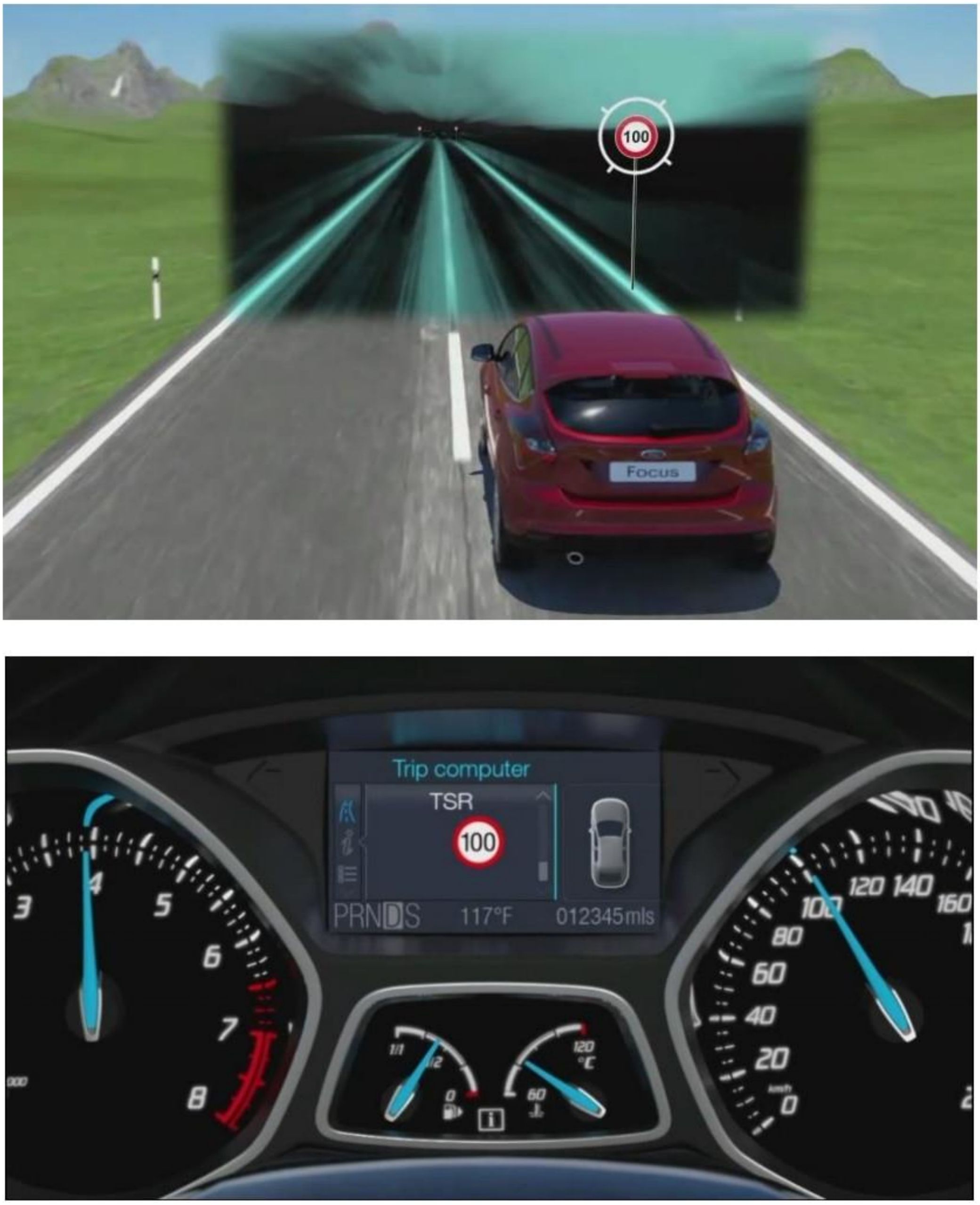

Figure 10. Traffic sign recognition system

CHAPTER 3
SYNTHESIS OF DETECTION AND RECOGNITION OF THE TRAFFIC SIGNS METHODS

3.1 Overview of the existing Detection and Recognition method

The developments on traffic sign detection and recognition methods have been under study for many years already. Since the appearance of the first paper in Japan in 1984, road sign recognition has become one of the important research fields. From that time until the present day many researchers, individual or groups, have been active in the field and have developed a high number of different methods and approaches to solving this problem.

In general, the identification of road signs can be carried out by two main stages: Traffic Signs Detection and Traffic Signs Recognition.

Traffic Signs Detection has some broad types, such as detection based on color segmentation, detection based on shape segmentation and detection based on the combination of two individual algorithms above.

Traffic Signs Recognition has an enormous amount of approaches. Recognition by using matching the information of the traffic signs with the existing templates. Besides that, Hough Transformation is used to detect the number of lines and peaks. Other methods are to use cross-correlation technique for recognizing the signs or using Support Vector Machines (SVM) to classify traffic sign shapes...

3.2 The Detection of Traffic signs

3.2.1 Color segmentation

3.2.1.1 RGB color space

In the RGB model, each color appears in its primary spectral components of red, green and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in Figure 11, in which RGB values are at three corners; cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the gray scale (point of equal RGB values) extends from the black to white along the line joining these two points. The different colors in this model are points on or inside the cube and are defined by vectors extending from the origin. For convenience, the assumption is that all color values have been normalized so that the cube shown in Figure 11 is the unit cube. That is all values of R, G, and B is assumed to be in the range [0, 1].

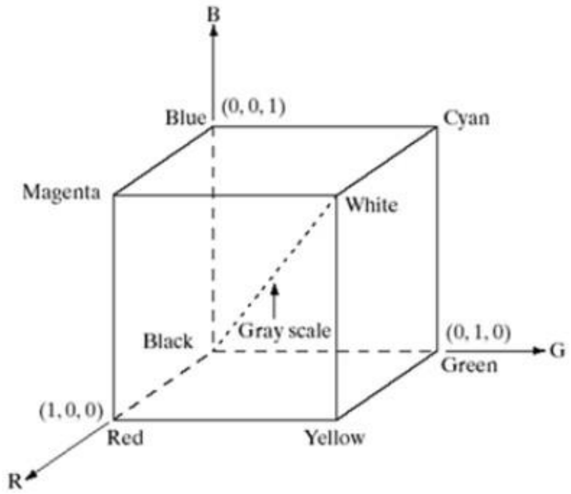

Figure 11. Schematic of the RGB color cube

The RGB color model is additive in the sense that the three light beams are added together, and their light spectra add the wavelength to wavelength, to make the final color's spectrum.

- Red + Green = Yellow
- Red + Blue = Purple
- Green + Blue = Cyan (blue-green)
- Red + Green + Blue = White

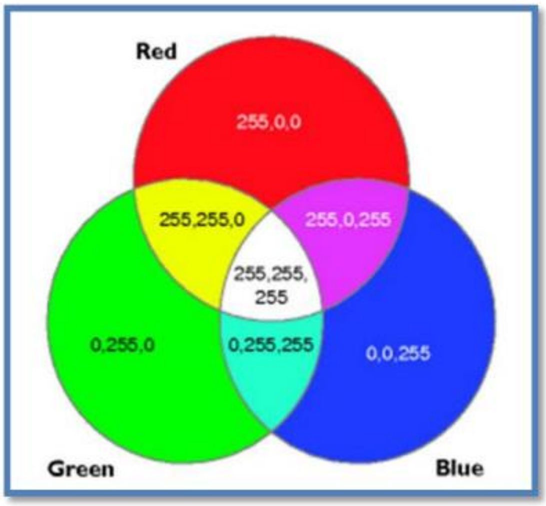

Figure 12. RGB color model

In color segmentation, the systems set up an image filter with RGB color space to extract three colors: red, white and black from the given image. Their color information will be stored respectively. The idea of this algorithm is to test if a candidate region is a speed sign by checking if there is a white region in a red region and if there is a black region in that white region [13].

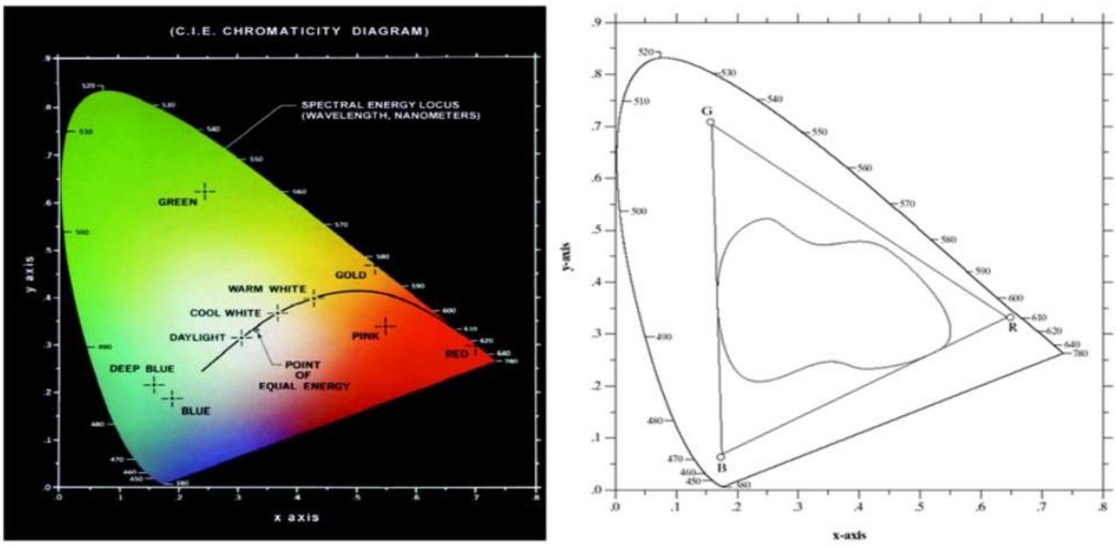

Figure 13. Color gamut produced by RGB monitors

Another way using RGB color in color segmentation is to convert from RGB image to grayscale image. They threshold the grayscale image to generate the binary image and then find out most likely white region as the candidate for further processes [14].

RGB color space is high sensitivity to the light condition. The segmentation result could vary in different light conditions due to weather and/or shadows. An appropriate fixed threshold value is also impossible to specify with this color space.

3.2.1.2 HSV color space

One of the most common color space in computer vision area is HSV, as well as in road sign detection and recognition field. This color space comprises three components: Hue, Saturation, and Value.

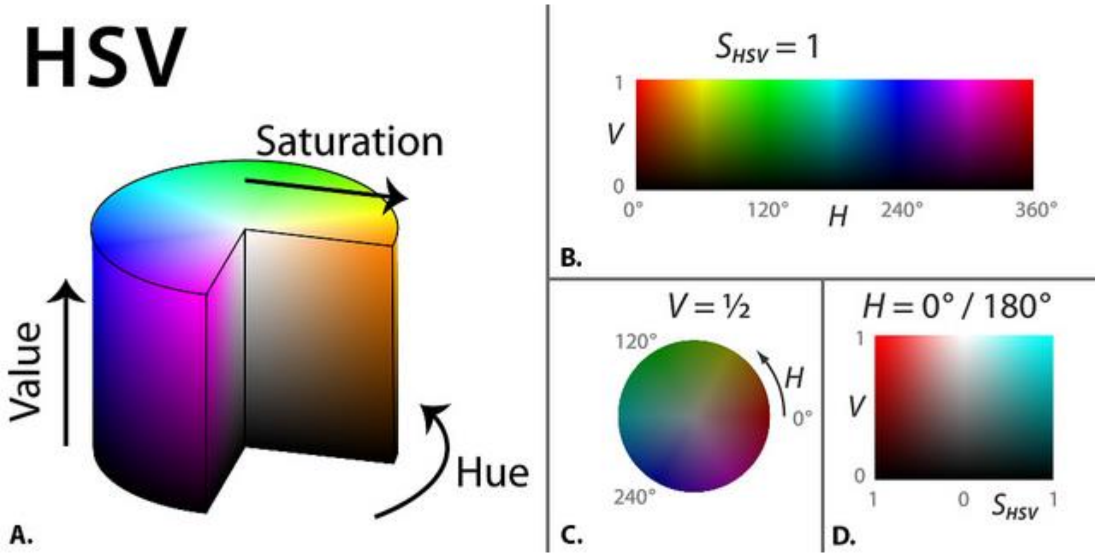

Figure 14. HSV Color Space

- Hue is defined color information.
  - 0 degrees is RED
  - 120 degrees is GREEN
  - 240 degrees is BLUE
- Saturation is defined as the range of gray in the color space.
  - Values range from 0 to 1.

- Value is defined brightness of the color. It varies with color saturation.
    - Values range from 0 to 1.

To segment color, the RGB original image is converted to HSV color space, then using a fixed threshold value extracts the color regions of interest. The threshold values are chosen by analyzing the hue, saturation and value information of the image [15]. Figure 15 shows that a good result of red color detection in HSV color space.

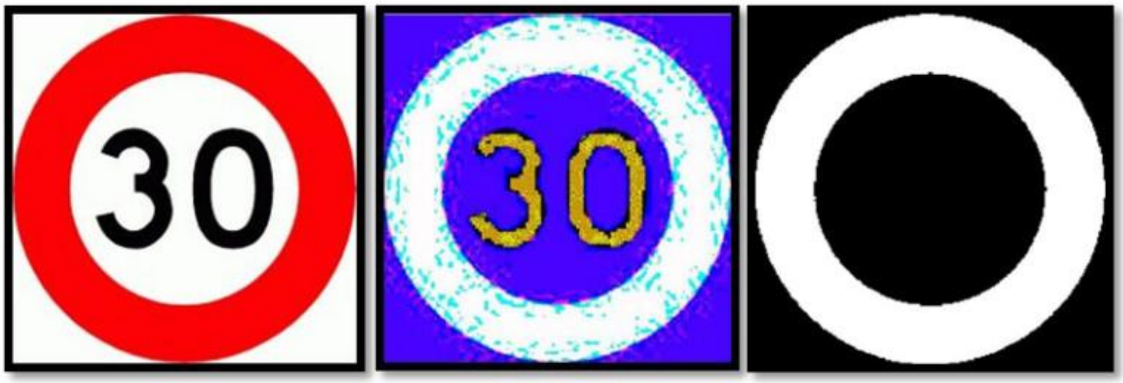

Figure 15. a. RGB original image. b. HSV image. c. Binary image extracts red color from HSV image.

The HSV color space has less variation to light condition than RGB. Hence, the effect of light can be reduced to an acceptable level; the fixed threshold value can be applied for segmentation.

3.2.1.3 Color segmentation in detection

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 [16] et al used Hue component for the sensory component to analyze the degree of similarity between the candidate image region and the road sign.

Shadeed et al. [17] proposed an algorithm to detect road signs using the HSV and YUV color spaces. The system is implemented in two stages. In the first stage, the

RGB image is converted into YUV color space, and a histogram of the Y channel is equalized, and then a new RGB is constructed. Color segmentation is achieved in the second stage by converting the RGB image generated by the first stage into HSV and YUV color spaces, and then applying a suitable value of threshold to H and UV values. Then the two results are combined by an AND operation.

H. Fleyeh [18] proposed a color segmentation method for traffic signs, namely the Shadow and Highlight Invariant Color Segmentation, which takes advantage of the invariant properties of the hue component.

Also process on RGB color space, the author in [19] found that ratios of RGB colors worked better and did not require color space conversion. The colors are segmented by compare between the ratios of red, green, blue, pair by pair, respectively with certain constants. These constants are determined by sampling the red, green, and blue values of images of typical signs. The percent of detection is 96%.

Influences of daily illumination changes are recognized by Benallal and Meunier [20]. Many experiments were carried out with several road signs to study the stability of colors under different illumination conditions. The RGB color space achieves segmentation. It is shown that differences between red and green and blue components respectively are high and could be used with an appropriate threshold for segmentation.

3.2.2 Shape recognition

To detect circle object, the system will calculate the circularity of the objects. Based on the properties of circle object, the circularity is computed by the formula:

$$\text{Circularity} = \frac{\text{perimeter}x^2}{4 \times \pi \times \text{area}}$$

According to Thomas B.Moeslund [21], be researched as:

Items	Circularity (C)

Circle	$0 < C \leq 1.2$
Square	$1.2 < C \leq 1.6$
Isosceles triangle	$1.6 < C \leq 1.8$
Others	$C > 1.8$

Table 1. The circularity values

a) Hough circles:

The Circular Hough Transform (CHT) is not a rigorously specified algorithm; rather some different approaches can be taken in its implementation. However, on the whole, there are three essential steps which are common to all [22].

- ✓ Accumulator Array Computation: Foreground pixels of the high gradient are designated as being candidate pixels and are allowed to cast ‘votes' in the accumulator array. In a standard CHT implementation, the candidate pixels vote in a pattern around them that forms a full circle of a fixed radius. Figure 16.a shows an example of a candidate pixel lying on an actual circle (solid circle) and the classical CHT voting pattern (dashed circles) for the candidate pixel.

- ✓ Center Estimation: The votes of candidate pixels belonging to an image circle tend to accumulate at the accumulator array bin corresponding to the circle's center. Therefore, the circle centers are estimated by detecting the peaks in the accumulator array. Figure 16.b shows an example of the candidate pixels (solid dots) lying on an actual circle (solid circle) and their voting patterns (dashed circles) which coincide at the center of the real circle.

- ✓ Radius Estimation: If the same accumulator array is used for more than one radius value, as is commonly done in CHT algorithms, radii of the detected circles have to be estimated as a separate step.

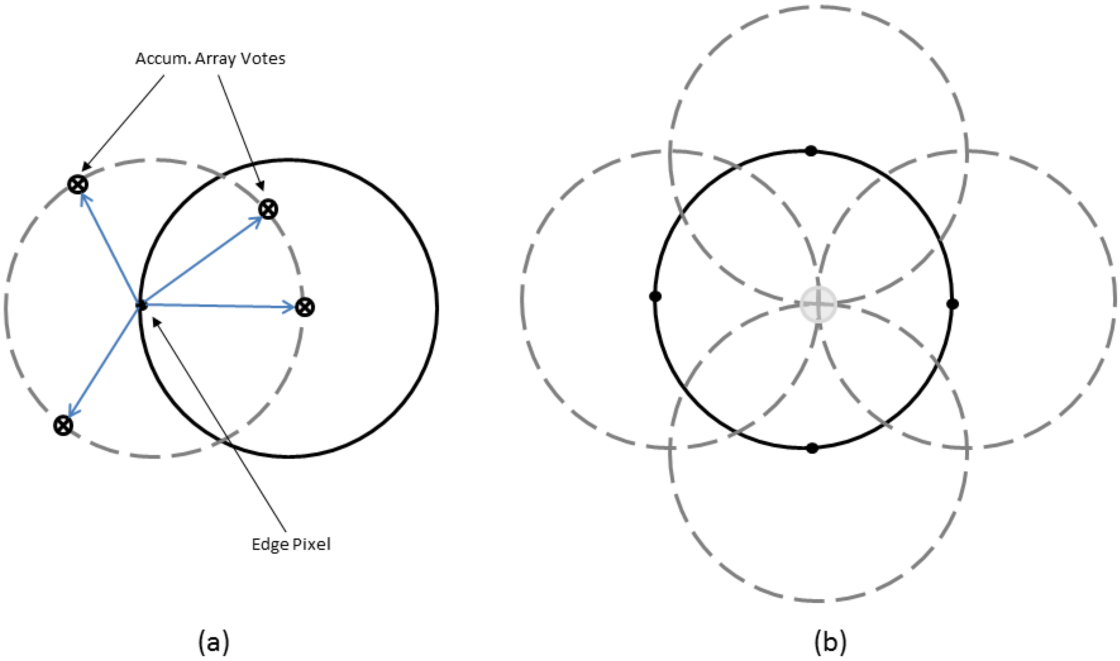

Figure 16. Classical CHT voting pattern

b) Affine Moment Invariant [23]

The standardized traffic signs have three distinct shapes: octagonal, triangular and circular. Therefore, the validation of the candidate regions is conducted to verify several shape measures such as ellipticity, triangularity, and rectangularity. Thus, a better solution is to evaluate the shape of the candidate regions using the Affine Moment Invariant (I_1):

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}{}^4}$$

Where:

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

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

To detect the object is a circle, we use the formula:

$$E = \begin{cases} 16\pi^2 I_1 & \text{if } \mathrm{I}_1 \le \frac{1}{16\pi^2} \\ \frac{1}{16\pi^2 I_1} & \text{otherwise} \end{cases}$$

The value of E is in the range [0,1] and an ellipticity value E = 1 corresponds to a completely circular shape.

To detect the object is a triangle, we use the formula:

$$T = \begin{cases} 108 I_1 & \text{if } \mathrm{I}_1 \le \frac{1}{108} \\ \frac{1}{108 I_1} & \text{otherwise} \end{cases}$$

The value of E is in the range [0,1] and a triangularity value T = 1 implies a perfectly triangular shape.

c) **Shape-arc detection** [24]

Shape-arc detection is an algorithm inspires by the real calculation of a simple object. For example, is a circle or a rectangle. Shape-arc detection consists of three major algorithms:

- ✓ Arc extraction from contours: The contours that will be extracted is the one that has been detected from the previous step (Ramer-Douglas-Peucker algorithm). An arc can be formed from three points. So the main idea of using the contour is to take for each three points which belongs to the same contour to make an arc.

- ✓ Arc grouping: grouping the arcs into one object. In this step, many algorithms can be used depend on the shape we want to detect. In the implementation, currently, only circle, rectangle, and triangle have been implemented.

- ✓ Shape estimation: Shape estimation can be combined with arc groupings like for rectangle or triangle. However, it can also be divided.

3.2.3 Traffic Sign Classifier (TSC) [25]

Traffic Sign Classifier based on the segmentation of Color Structure Code (CSC). All color classes relate to the color of the real traffic signs, e.g. red, blue, yellow, white. The ranges of our color classes are well defined for various lighting conditions caused by shadows or the varying color temperature of daylight. The next step is a geometric classification of the color objects. Therefore, for every object, we compute the probability that it belongs to a certain shape class. The shape classes depend on the different colored traffic signs, e.g. circles, semicircles, triangles, rectangles. Its convex hull encodes the shape of an object. TSC detects the classes of traffic signs which are described by its color and shape and if it does not belong to classes, it will be eliminated. This method has high performance and can run in real-time, but it requires a big training data and much time for training.

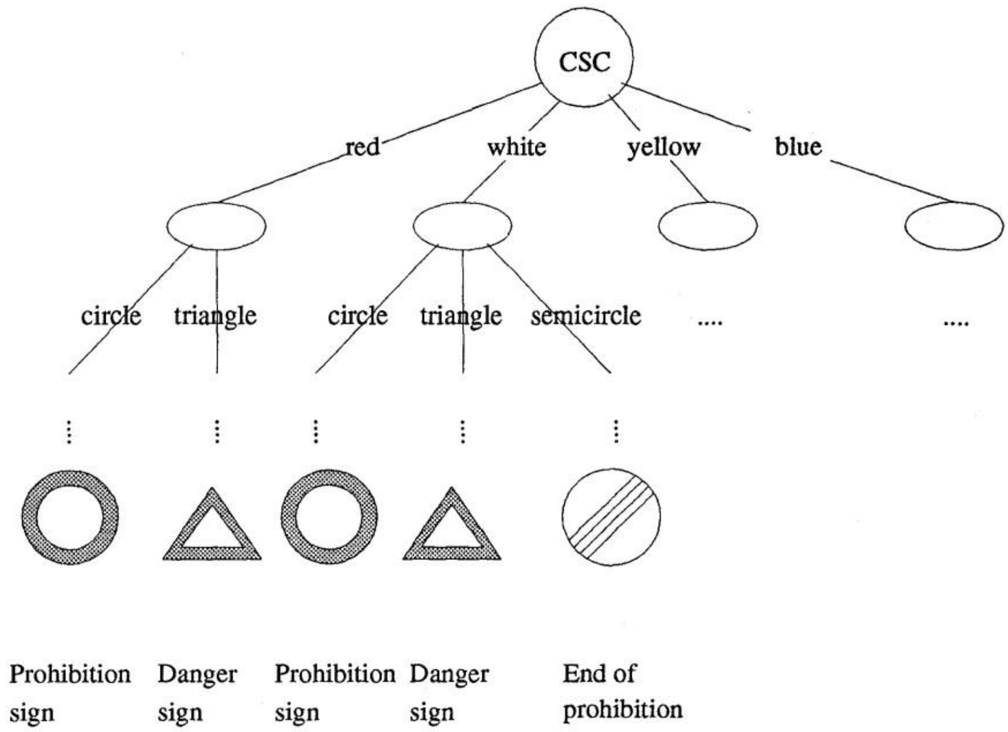

Figure 17. The TSC decision tree

3.3 The Recognition of Traffic Signs

Once the region of interest is determined, and a traffic sign is detected, it should be recognized using a predetermined database of all the traffic signs in the system.

Claw Bahlmann et al [26] proposed an approach to detect traffic signs by using a set of Haar wavelet features obtained from AdaBoost training, joint modeling of color and shape information without the need of tuning free parameters.

Ohara et al. [27] used template matching for sign recognition. A sub-area of size NxN is selected, and small defects and noises are filled in or deleted. The pixel values are normalized by the maximum and minimum values of the input sub-area. The size is also normalized depending on the template to be matched, and the closeness with the template is calculated. A recognition rate of over 95% is achieved.

Lafuente-Arroyo et al. [28] used Support Vector Machines (SVM) to classify traffic sign shapes based on the distance of the edge of the sign from the border of the normalized image. The method achieved 82% success rate for standalone signs and 54% for occluded signs.

Another approach using SIFT (Scale Invariant Feature Transform) for recognition traffic sign. It first creates scale space; then key points are detected using 16 DoG image. Then SIFT descriptors are extracted using gradient magnitude and orientation. This approach resulted in 91.1% in recognition rate [29].

3.3.1 Introduction to Support Vector Machine (SVM)

The original SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963. In 1992, Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes. The current standard incarnation (soft margin) was proposed by Corinna Cortes and Vapnik in 1993 and published in 1995 [30].

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

3.3.1.1 Linearly Separable Data [31]:

Consider the image below which has two types of data, red and blue. For a test data, we used to measure its distance to all the training samples and take the one with minimum distance. It takes plenty of time to measure all the distances and plenty of memory to store all the training samples.

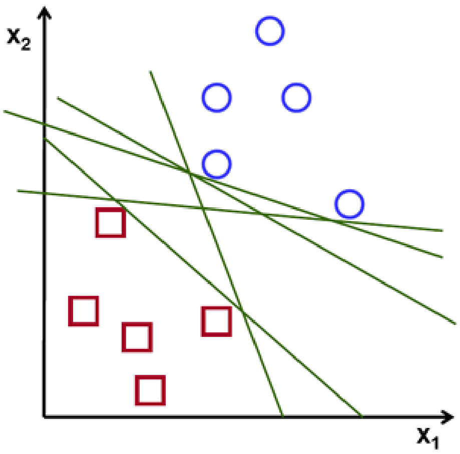

Figure 18. Multiple lines in the hyperplane

Consider another idea. We find a line, $f(x) = ax_1 + bx_2 + c$ which divides both the data to two regions. When we get a new test data X, just substitute it in f(x). If f(X) > 0, it belongs to a blue group, else it belongs to the red group. We can call this line as **Decision Boundary**. It is very simple and memory-efficient. Such data which can be divided into two with a straight line (or hyperplanes in higher dimensions) is called **Linear Separable**.

So in above image, you can see plenty of such lines are possible. Which one will we take? Very intuitively we can say that the line should be passing as far as possible from all the points. Why? Because there can be noise in the incoming data. This information should not affect the classification accuracy. So taking a farthest line will provide more immunity against noise. So what SVM does is to find a straight line (or hyperplane) with largest minimum distance to the training samples. See the bold line in below image passing through the center.

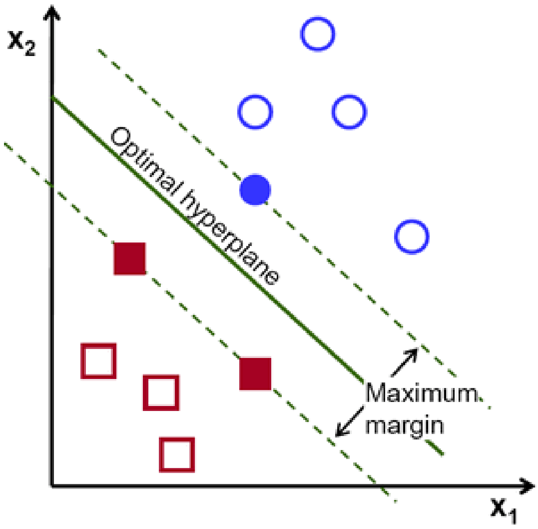

Figure 19. The optimal hyperplane

So to find this Decision Boundary, you need training data. Do you need all? NO. Just the ones which are close to the different group are sufficient. In our image, they are the one blue filled circle and two red filled squares. We can call them **Support Vectors** and the lines passing through them are called **Support Planes**. They are adequate for finding our decision boundary. We need not worry about all the data. It helps in data reduction.

What happened is, first two hyperplanes are found which best represents the data. For, e.g., blue data is represented by $w^T x + b_0 > 1$ while red data is represented by where:

w is $w^T x + b_0 < -1$

weight vector is $w = [\, w_1, w_1, \ldots, w_n]$

x is the feature vector $x = [\, x_1, x_1, \ldots, x_n]$

b_0 is the **bias**.

Weight vector decides the orientation of decision boundary while bias point decides its location. Now decision boundary is defined to be midway between these hyperplanes so expressed as $w^T x + b_0 = 0$. The minimum distance from support vector to the decision boundary is given by $distance_{supportmvectors} = \frac{1}{\|w\|}$, Margin is twice this distance, and we need to maximize this margin. i.e. we need to minimize a new function $L(w, b_0)$ with some constraints which can express below:

$$\min_{w,b_0} L(w, b_0) = \frac{1}{2}\|w\|^2 \; subject \; to \; t_i(w^T x + b_0) \geq 1 \; \forall i$$

where t_i is the label of each class, $t_i \in [-1, 1]$

3.3.1.2 Non-Linearly Separable Data [31]:

Consider some data which can't be divided into two with a straight line. For example, consider a one-dimensional data where 'X' is at -3 & +3 and 'O' is at -1 & +1. Clearly it is not linearly separable. However, there are methods to solve these kinds of problems. If we can map this data set with a functi $f(x) = x^2$on, we get 'X' at 9 and 'O' at one which are linearly separable.

Otherwise, we can convert this one-dimensional to two-dimensional data. We can use $f(x) = (x, x^2)$ a function to map this data. Then 'X' becomes (-3,9) and (3,9) while 'O' becomes (-1,1) and (1,1). This is also linear separable. In short, the chance is more for a non-linear separable data in lower-dimensional space to become linear separable in higher-dimensional space.

In general, it is possible to map points in a d-dimensional space to some D-dimensional space $(D > d)$ to check the possibility of linear separability. There is an idea which helps to compute the dot product in the high-dimensional (kernel) space by performing computations in the low-dimensional input (feature) space. We can illustrate with the following example.

Consider two points in two-dimensional space, $p = (p_1, p_2)$ and $q = (q_1, q_2)$. Let $\emptyset$ be a mapping function which maps a two-dimensional point to three-dimensional space as follows:

$$\emptyset(p) = \left(p_1^2, p_2^2, \sqrt{2}p_1p_2\right) \qquad \emptyset(q) = \left(q_1^2, q_2^2, \sqrt{2}q_1q_2\right)$$

Let us define a kernel function $K(p, q)$ which does a dot product between two points, shown below:

$$\begin{aligned} K(p,q) = \emptyset(p).\emptyset(q) &= \emptyset(p)^T, \emptyset(q) \\ &= \left(p_1^2, p_2^2, \sqrt{2}p_1p_2\right).\left(q_1^2, q_2^2, \sqrt{2}q_1q_2\right) \\ &= {p_1}^2{q_1}^2 + {p_2}^2{q_2}^2 + 2p_1q_1p_2q_2 \\ &= (p_1q_1 + p_2q_2)^2 \\ \emptyset(p).\emptyset(q) &= (p.q)^2 \end{aligned}$$

It means, a dot product in three-dimensional space can be achieved using squared dot product in two-dimensional space. This can be applied to higher dimensional space. So we can calculate higher dimensional features from lower dimensions itself. Once we map them, we get a higher dimensional space.

In addition to all these concepts, there comes the problem of misclassification. So just finding decision boundary with maximum margin is not sufficient. We need to consider the issue of misclassification errors also. Sometimes, it may be possible to find a decision boundary with less margin, but with reduced misclassification. Anyway, we need to modify our model such that it should find decision boundary with maximum margin, but with less misclassification. The minimization criteria are amended as:

$$\min\|w\|^2 + C(distance\ of\ misclassified\ samples\ to\ their\ correct\ regions)$$

Below image shows this concept. For each sample of the training data a new parameter, ξ_i is defined. It is the distance from its corresponding training sample to their correct decision region. For those who are not misclassified, they fall on their corresponding support planes, so their distance is zero.

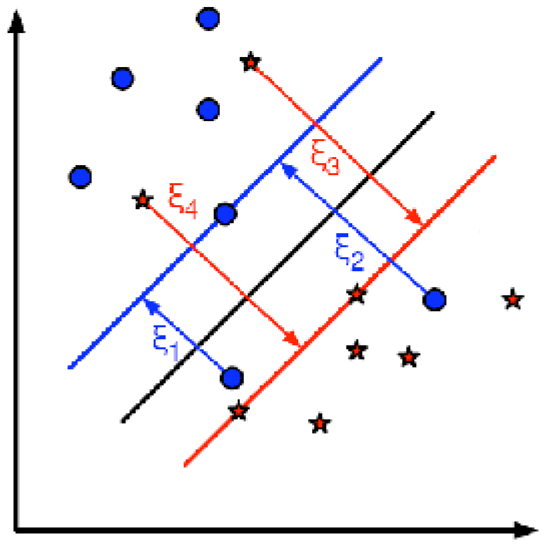

Figure 20. The relevant support planes

So the new optimization problem is:

$$\min_{w,b_0} L(w,b_0) = \|w\|^2 + C\sum_i \xi_i \; subject\ to\ y_i(w^T x + b_0) \geq 1 - \xi_i \; and\ \xi_i \geq 0\ \forall i$$

How should the parameter C be chosen? It is evident that the answer to this question depends on how the training data is distributed. Although there is no general answer, it is useful to take into account these rules:

Large values of C give solutions with fewer misclassification errors but a smaller margin. Consider that, in this case; it is expensive to make misclassification errors. Since the aim of the optimization is to minimize the argument, few misclassifications errors are allowed.

Small values of C give solutions with bigger margin and more classification errors. In this case, the minimization does not consider that much the term of the sum, so it focuses more on finding a hyperplane with a significant margin.

3.3.2 Introduction to Artificial neural network (ANN)

ANN is information processing model is modeled on the operation of the nervous system of the organism, including a large number of Neuron is mounted to process information. ANN is the same as the human brain, which is learned by experience (through training), capable of keeping the learning experience (knowledge) and use that knowledge to predict the unknown data (unseen data).

ANN's applications are used in many areas such as electricity, electronics, economics, military to solve the problem of complexity and requires high precision automatic control dynamic, identity...

3.3.2.1 General architecture of an ANN

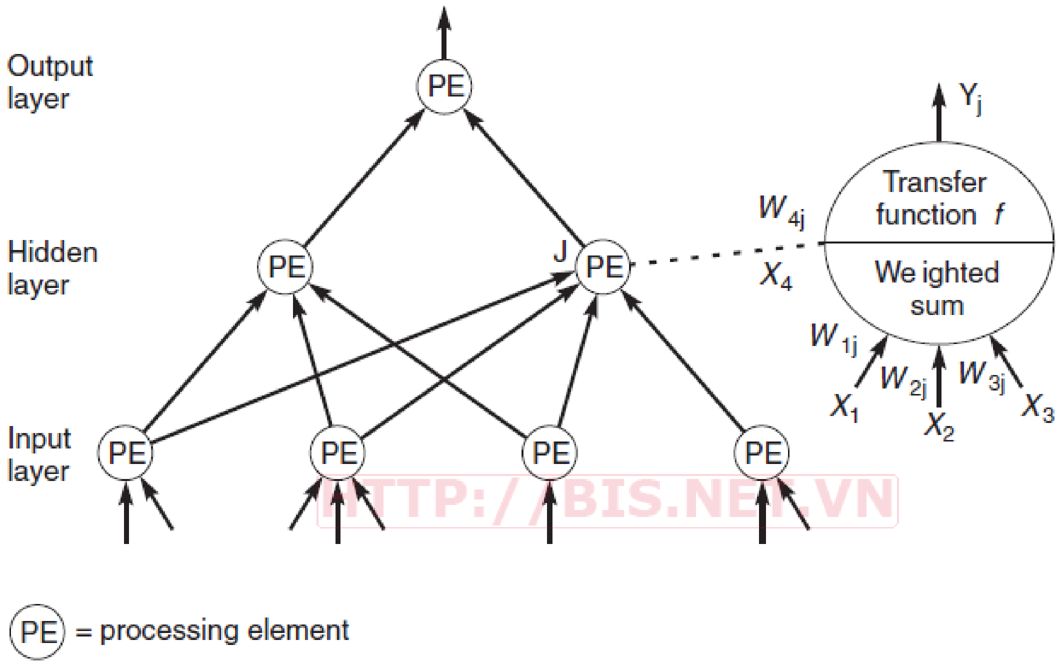

Figure 21. General architecture of an ANN

Processing Elements (PE): The PE of ANN known as Neuron, each Neuron receive Inputs process them and produce a result (Output) only. Results of a Neuron processor can do Inputs for other Neuron.

The general architecture of ANN has three components: **Input Layer**, **Hidden Layer,** and **Output Layer** (see the above Figure).

In which, the Hidden Layer consists of Neuron, receive data input from the previous neuron Layer and convert this input to the next processing layer. In an ANN can have multiple Hidden Layer.

Information processing of an ANN

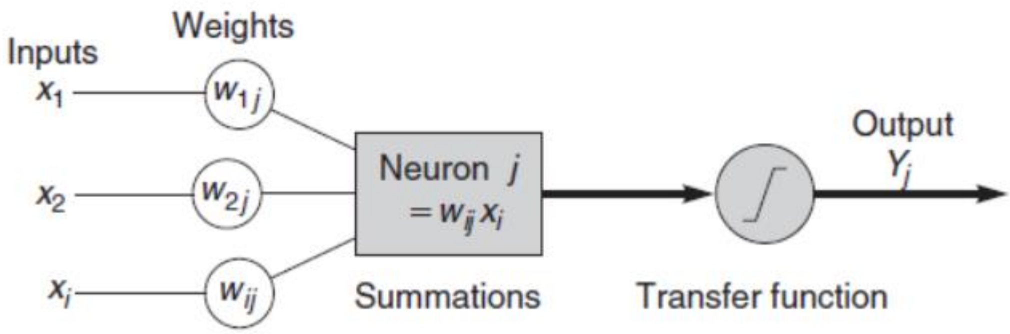

Figure 22. The processing of an ANN

Inputs: Each Inputs associated with one attribute of patterns. For example, in the application of the bank, it will consider acceptable for customers to borrow money or not, so each Inputs is a customer's attributes such as income, occupation, age, the number of children and so on.

Output: The result of an ANN is a solution to a problem, such as the issue of the consideration for the customer to accept a loan or not, Output is yes (lend) or no (not lend).

Connection Weights: This is a critical component of an ANN. It represents a significant level (intensity) of the input data for the processing of information (data conversion process from layer to layer other). The Learning Processing of ANN is the process of adjusting the **Weights** of the input data to obtain the desired results.

Summation Function: Calculate the weighted sum of all inputs are included in each neuron (**PE**). General function of a neuron for n **Inputs** is calculated according to the following formula:

$$Y = \sum_{i=1}^{n} X_i W_i$$

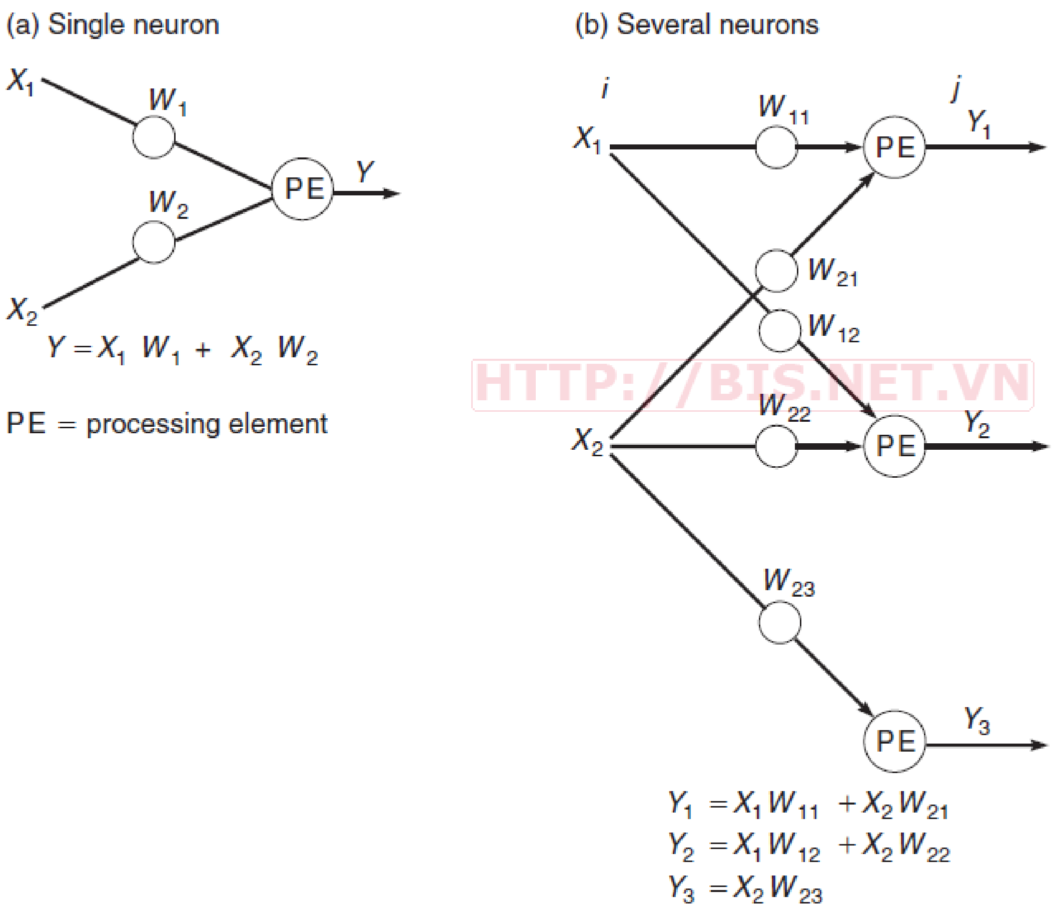

Figure 23. Summation function for single and several neurons

Transformation Function: Summation Function of a Neuron show the Activation of Neuron is called the Internal Activation. The Neuron can generate an output or not in ANN (in other words that, the Output of a Neuron can be transferred to the next layer of ANN or not). The Transfer Function represents the relationship between the Internal Activation and Output.

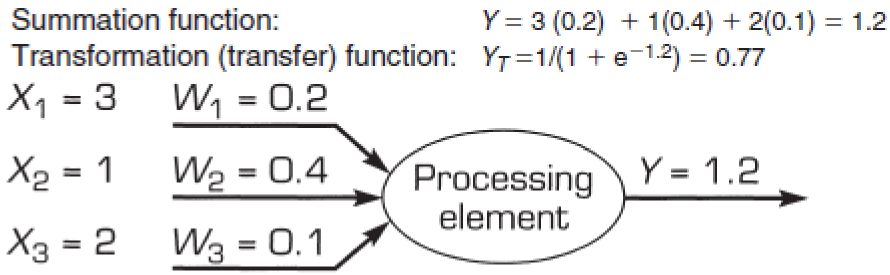

Figure 24. Transformation Function of a Neuron

The choice of Transformation function has a major impact on the results of ANN. The nonlinear conversion function is used commonly in the ANN is the Sigmoid (logical activation) function.

$$Y_T = \frac{1}{1 + e^{-Y}}$$

where:

Y_T: is the Transformation function

Y: is the Summation function

The results of the Sigmoid function ranges of the interval [0,1], so it is called the Normalized Function.

The results processed in the Neuron (Output) is sometimes enormous, so the Transformation function is used to handle this Output before transferring to the next layer. Instead of using Transfer function, people sometimes use a Threshold value to control the output of the Neurons in a certain layer of this output before transferring to the next tier. If the output of a given Neuron is smaller than the Threshold, it will not be moved to the next tier.

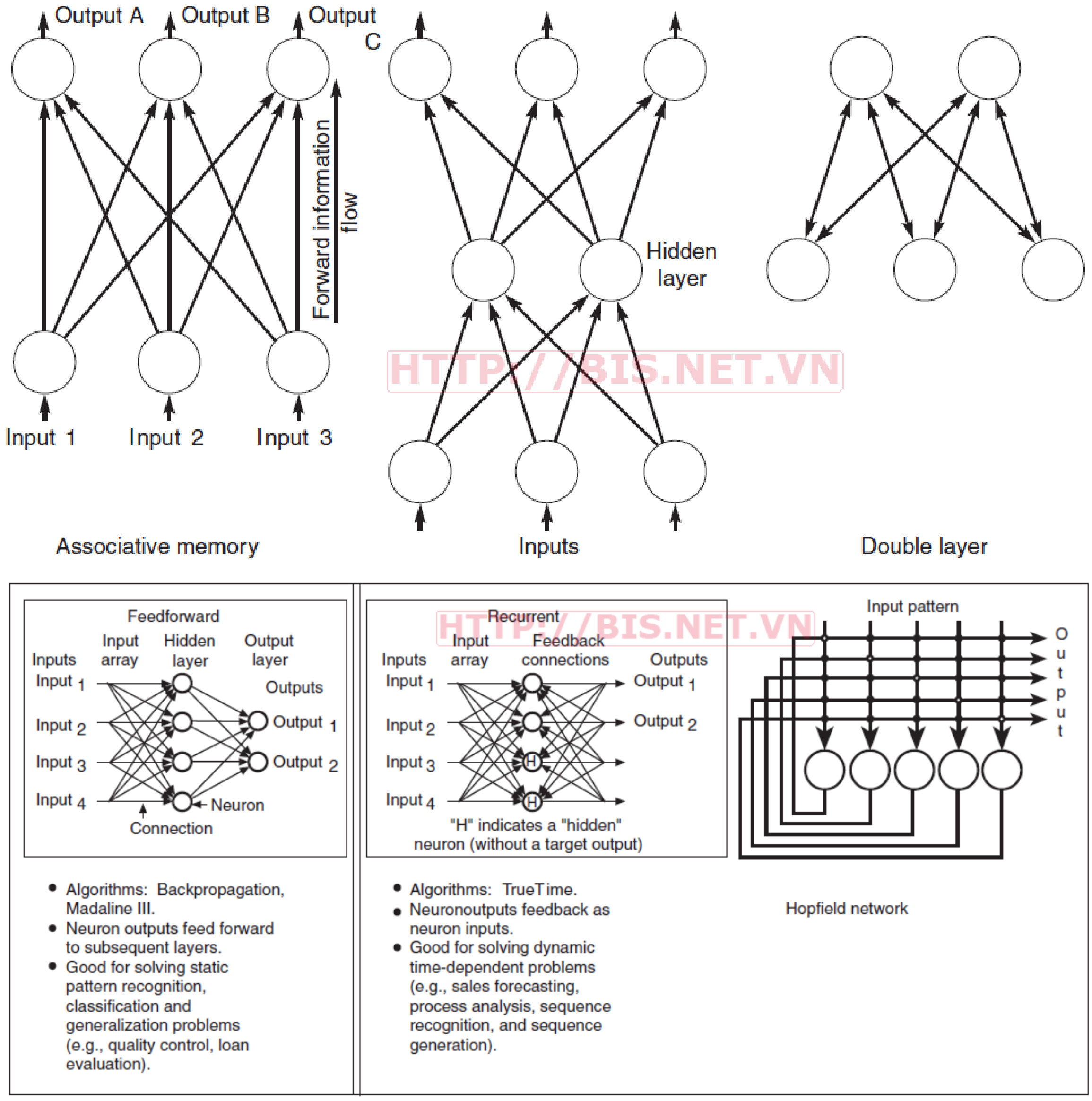

Figure 25. Some architectures of ANN

3.3.2.2 Learning Processing of ANN

ANN is trained or learned under two basic techniques that are Supervised Learning and Unsupervised Learning.

Supervised learning: The training process is repeated until the result (Output) of the ANN achieve the Desired value. Typically for this technique is the Backpropagation.

Unsupervised learning: Do not use external knowledge in the Learning process, so it is called Self - Organizing. Neuron networks are typically training by unsupervised style is Self - Organizing Map (SOM).

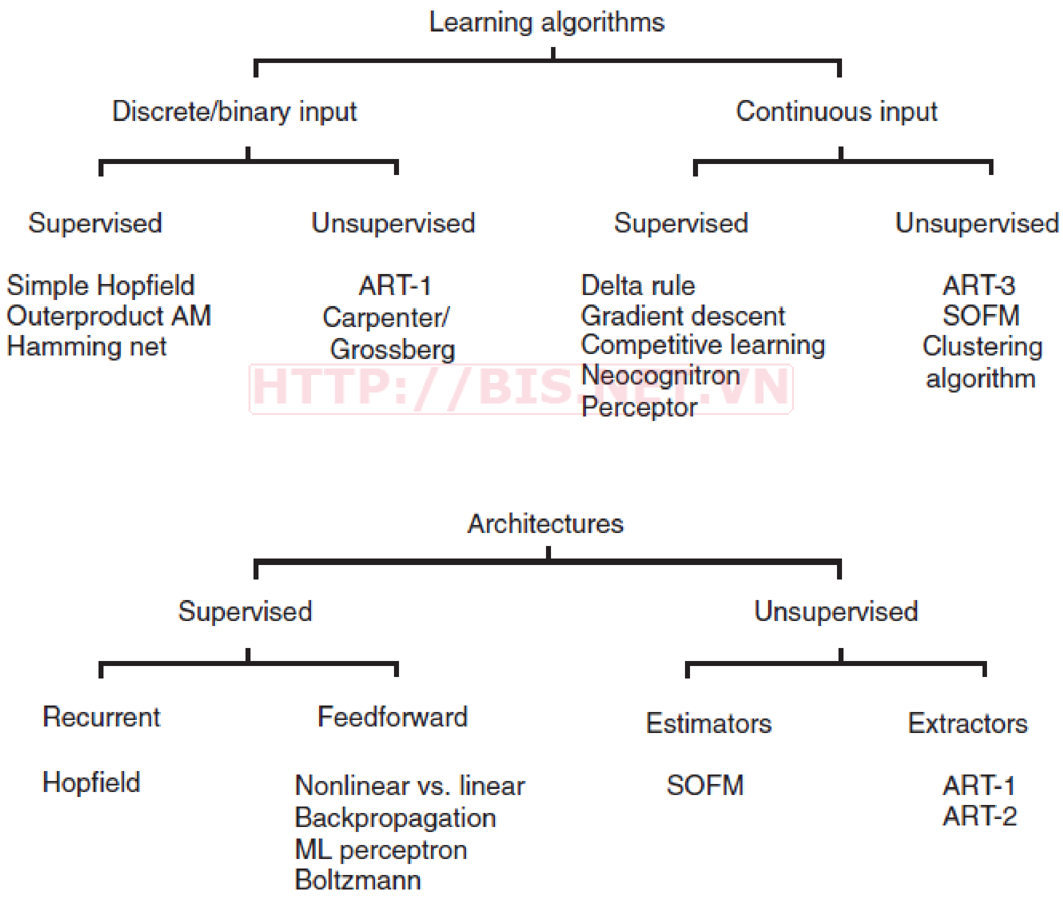

Figure 26. The algorithms and architectures of ANN Learning

CHAPTER 4

IMPROVED DETECTION AND RECOGNITION OF THE TRAFFIC SIGNS METHODS IN NIGHTTIME

After analysis and research the related algorithms to the problem of detecting and recognizing traffic signs, the Figure 27 below is the model to identify and recognize traffic signs. This model has six stages: Input tier, Pre-process tier, Color segmentation tier, Detection tier, Recognition tier, and Result tier.

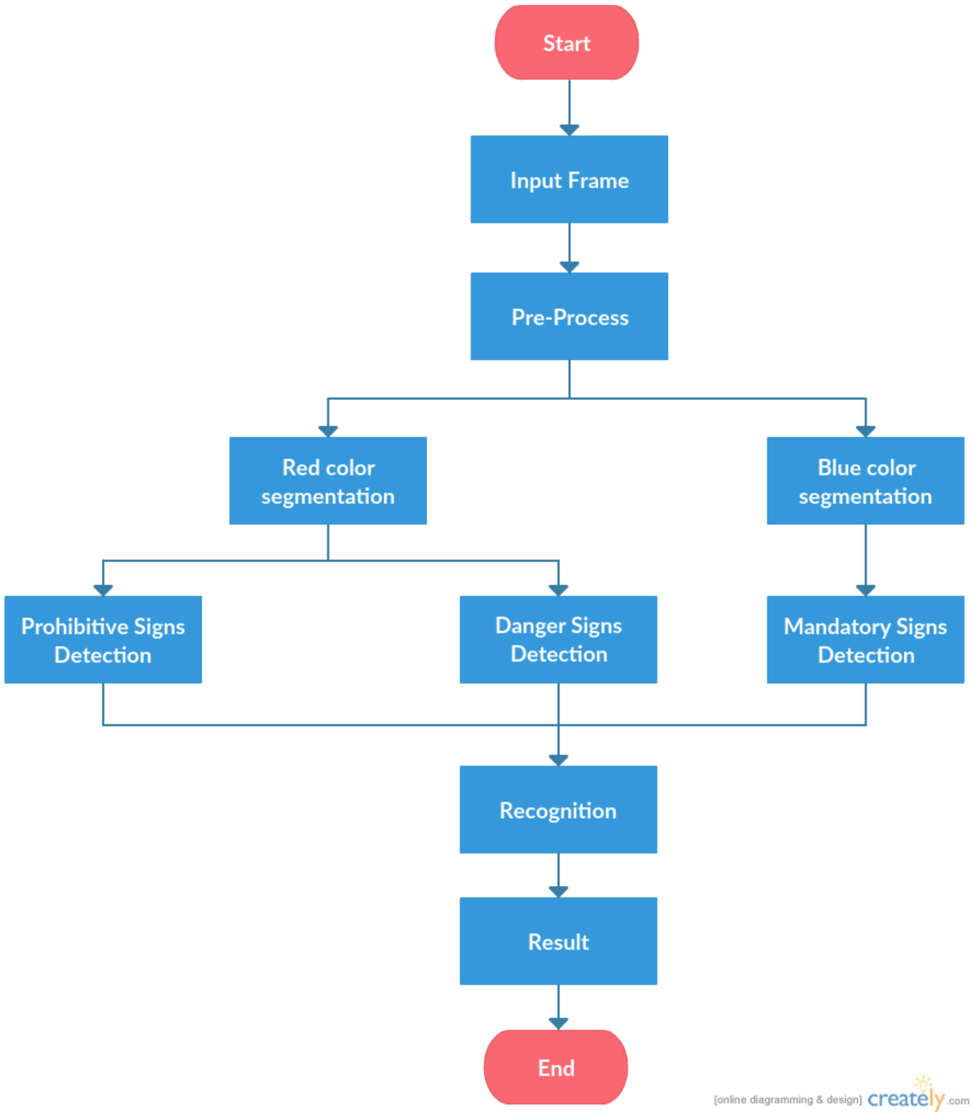

Figure 27. Proposed method for Traffic Signs Detection and Recognition

Below are two Proposed methods for the Detection and Recognition of Traffic Signs system: Traffic Signs Detection and Traffic Signs Recognition.

4.1 Method for Traffic Signs Detection

4.1.1 Pre-process

The images obtained on the road is always influenced by light conditions or camera quality. The object is achieved at a different distance and environmental conditions, so it is very difficult to distinguish the object in the image. Therefore, the Pre-process step will be applied to improve the image quality. The Figure 28 below is the model of Pre-process phase.

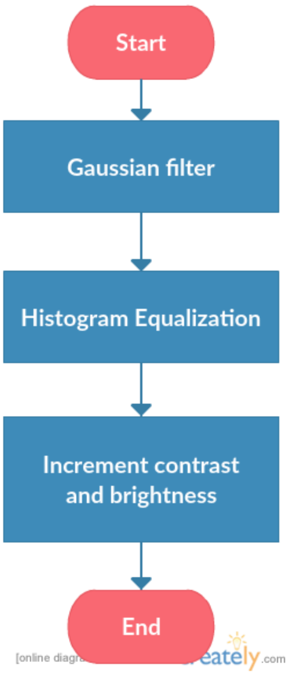

Figure 28. Pre-process model

4.1.1.1 Gaussian filter

Gaussian filter is applied to remove noise. It will try to smooth or blur the image without losing its edges. Gaussian filtering is done by convolution each point in the input image with a Gaussian kernel and then aggregating all of them to produce the output image. The example is shown in Figure 29.

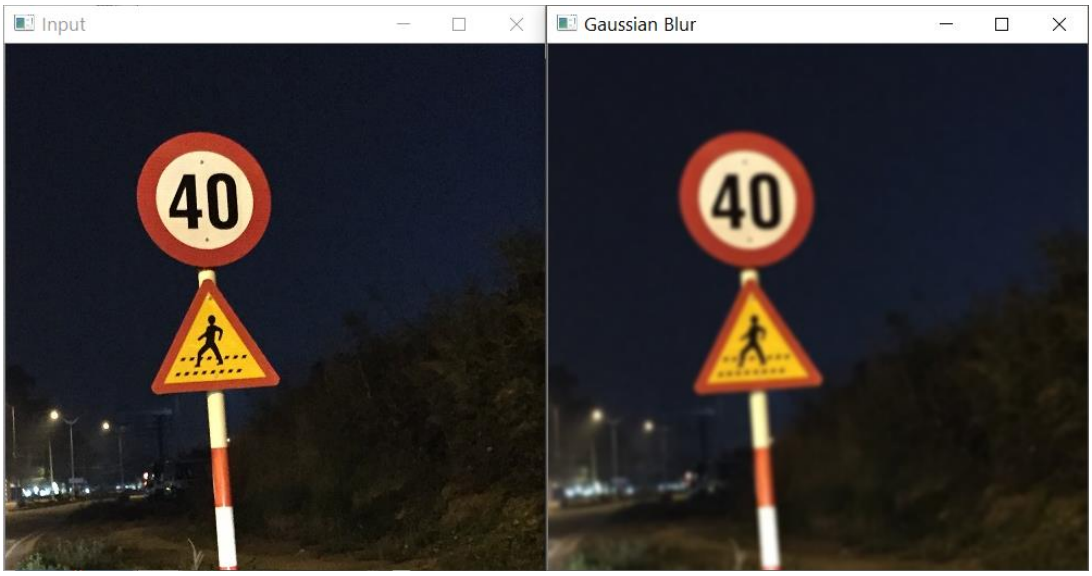

Figure 29. Gaussian filter

4.1.1.2 Histogram Equalization

Histogram equalization is used to improve the contrast and correct color distribution. In this project, I apply the Histogram equalization with CLAHE method (Contrast Limited Adaptive Histogram Equalization), it splits the space in a grid and applies the equilibrium to each tile. The example is shown in Figure 30.

Figure 30. Histogram equalization with CLAHE method

4.1.1.3 Increase contrast and brightness

Increasing contrast and brightness will make the image clearer and remove low contrast signals, which rising the ability to identify signs in the next stage. The example is shown in Figure 31.

Figure 31. Adjusting the contrast and brightness

4.1.2 Red color segmentation

After the Pre-process phase, the imageThe Figure 32 show the algorithm to detect the Danger and Prohibitory signs.

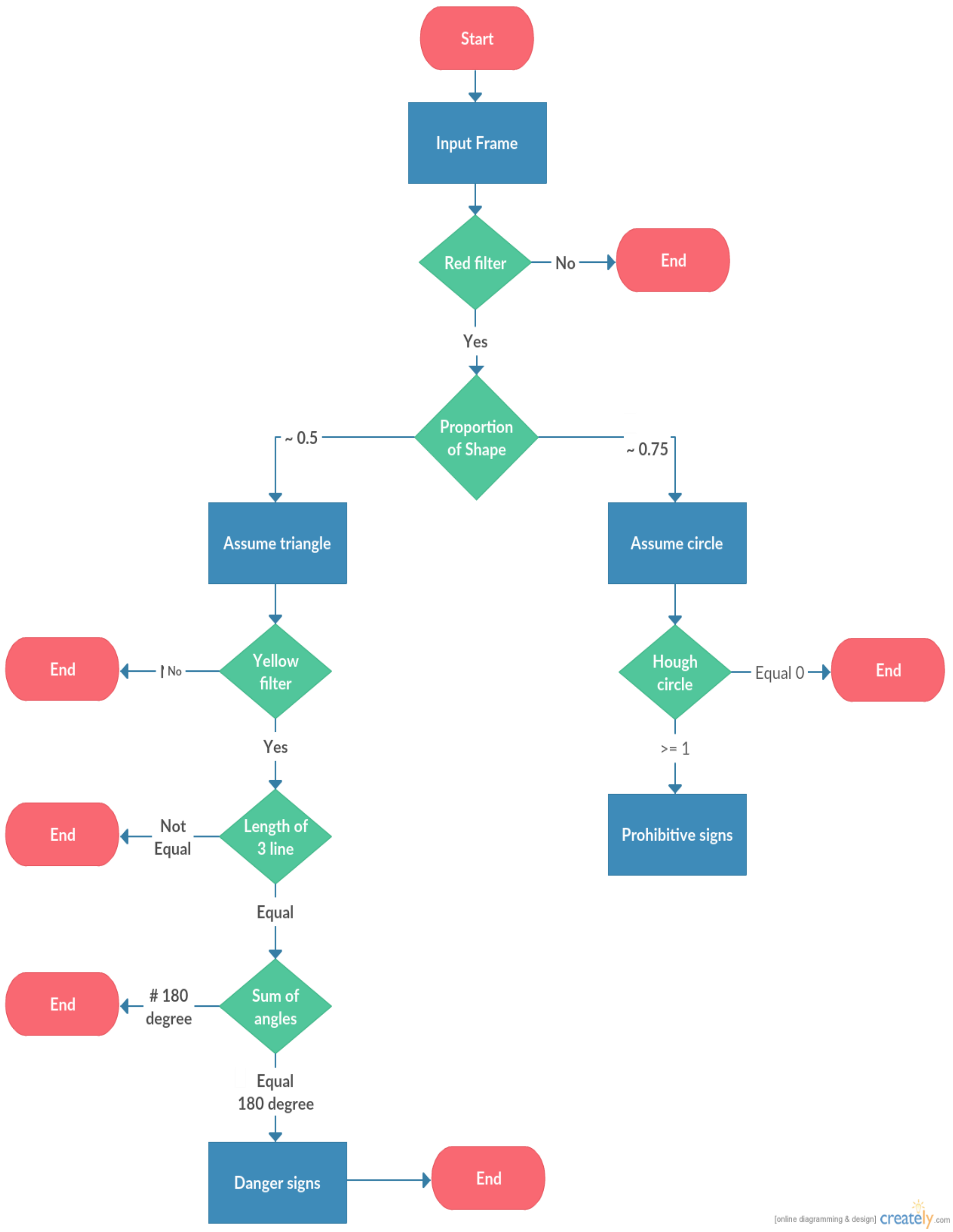

Figure 32. Danger and Prohibitory Signs Detection

This is the result after detecting the Danger and Prohibitory signs. The signs will mark with the bounding rectangle.

	Circle	Triangle	Combine	Figure
Daytime	x			Figure 33
		x		Figure 34
		x		Figure 35
			x	Figure 36
Nighttime	x			Figure 37
			x	Figure 38

Table 2. The result of Detection phase in Red color segmentation

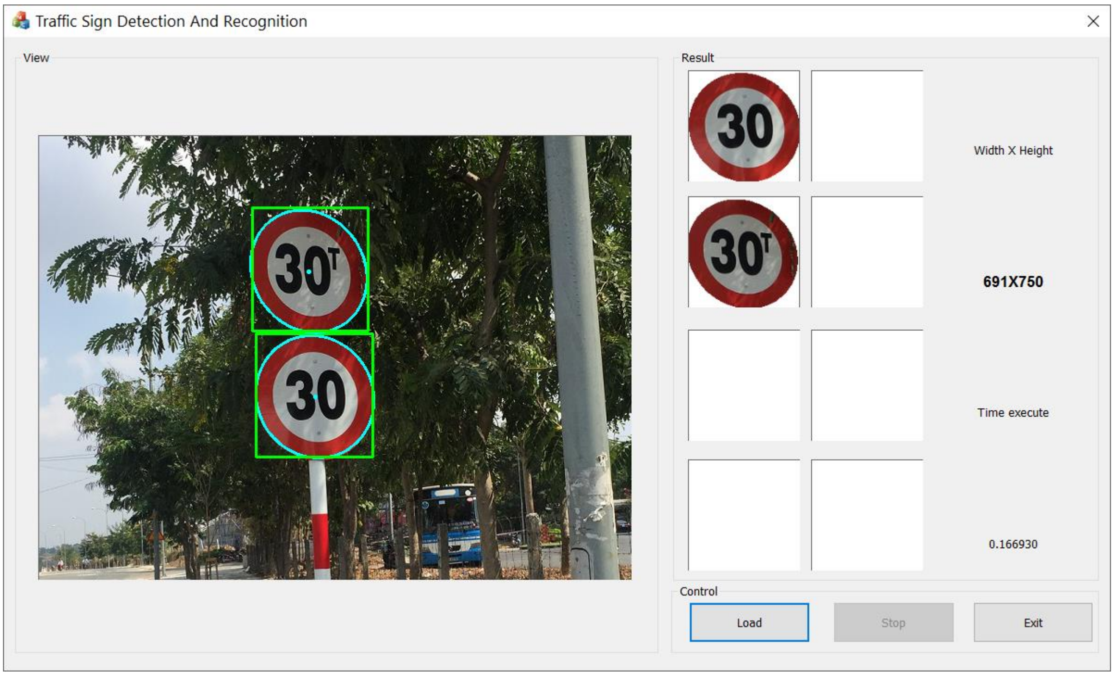

Figure 33. The two mutually occluding Prohibitory Signs

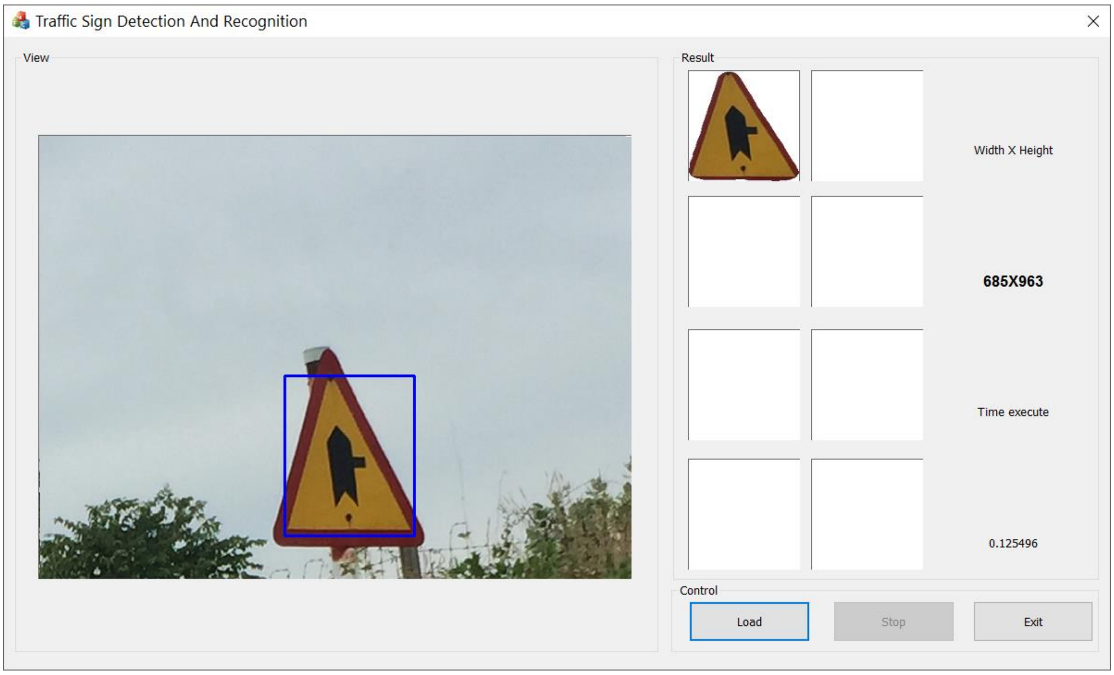

Figure 34. The Skewed Danger Signs

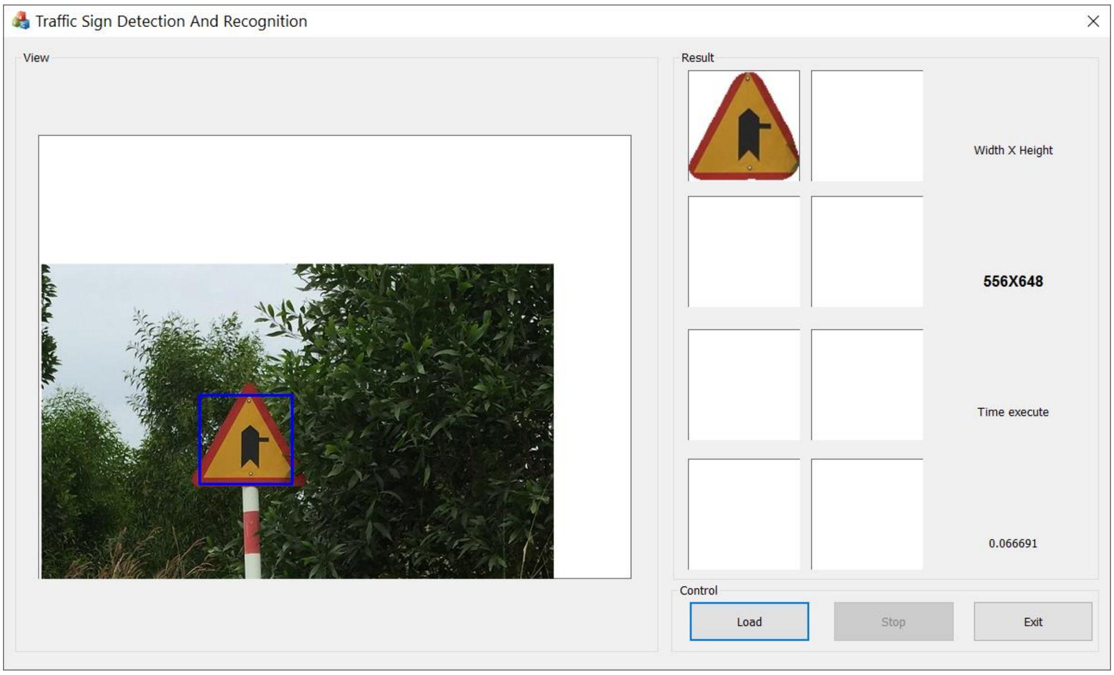

Figure 35. The Obscured part Danger Signs

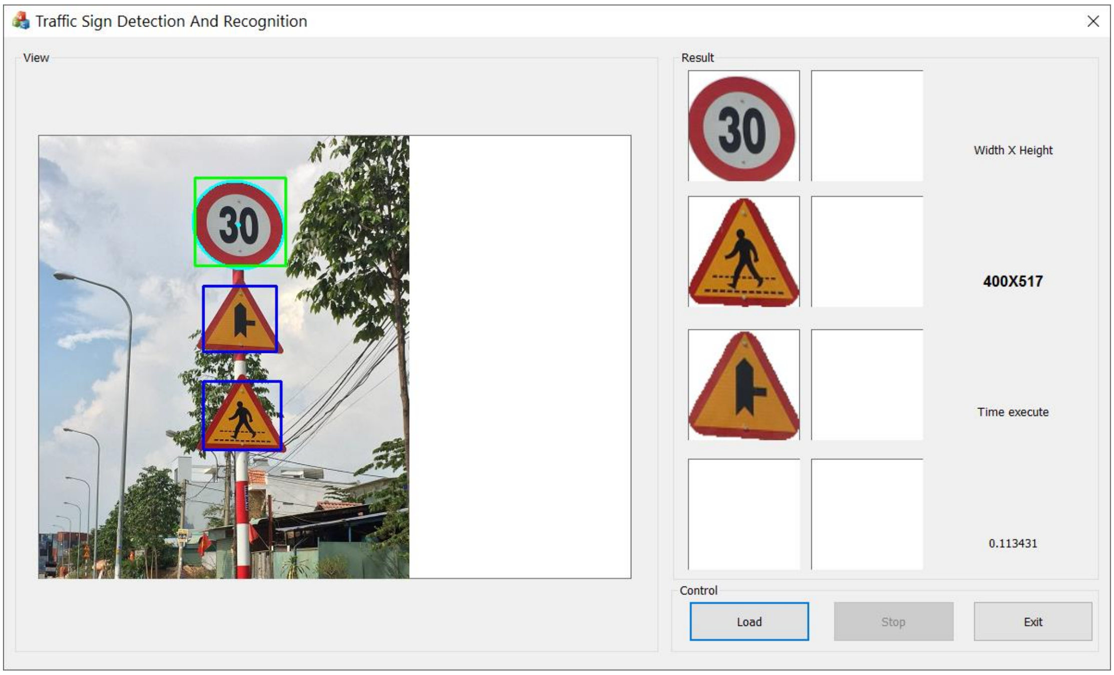

Figure 36. The three mutually occluding Prohibitory and Danger Signs

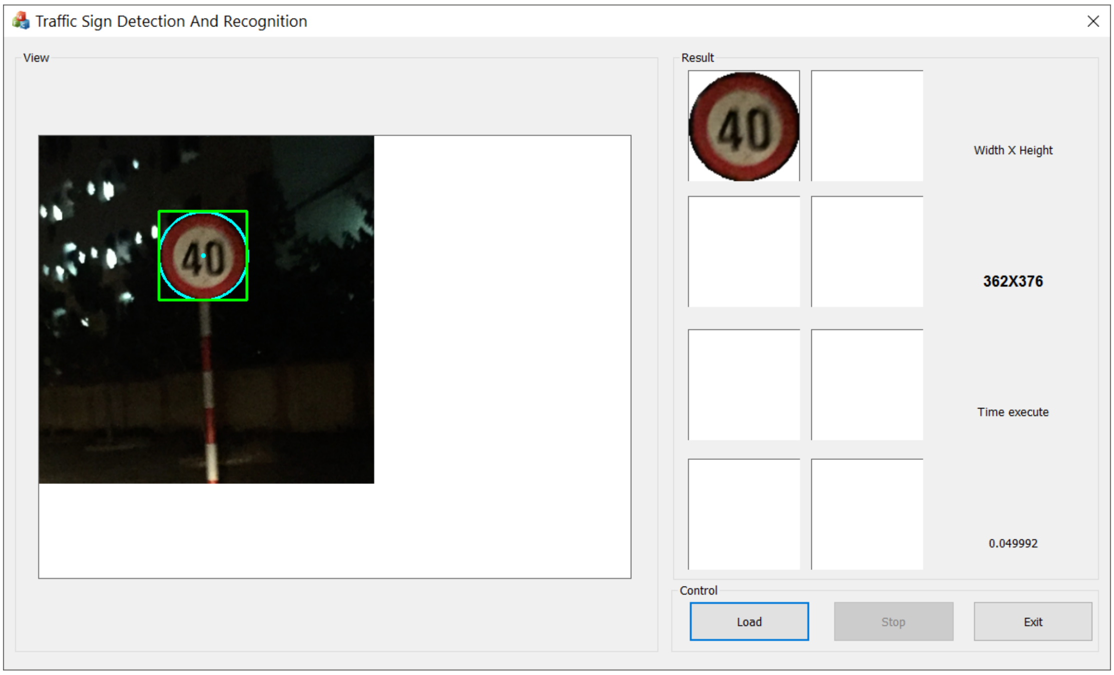

Figure 37. The Prohibitory Signs in Nighttime

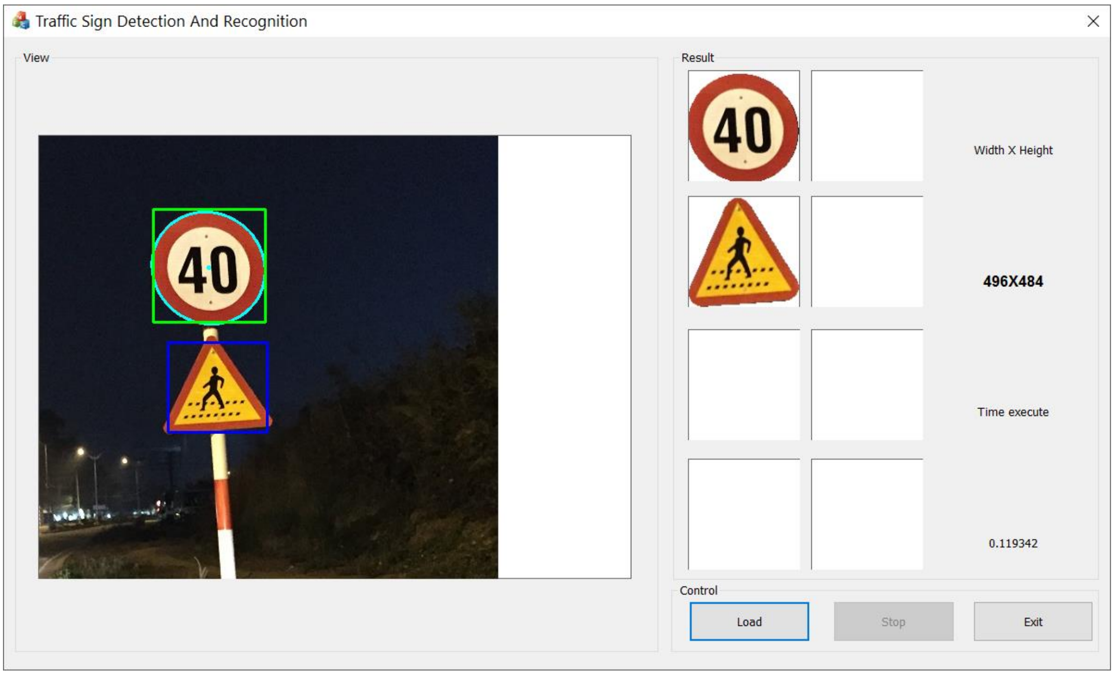

Figure 38. The Prohibitory and Danger Signs in Nighttime

4.1.3 Blue color segmentation

The Figure 39 will show the algorithm to detect the Mandatory signs.

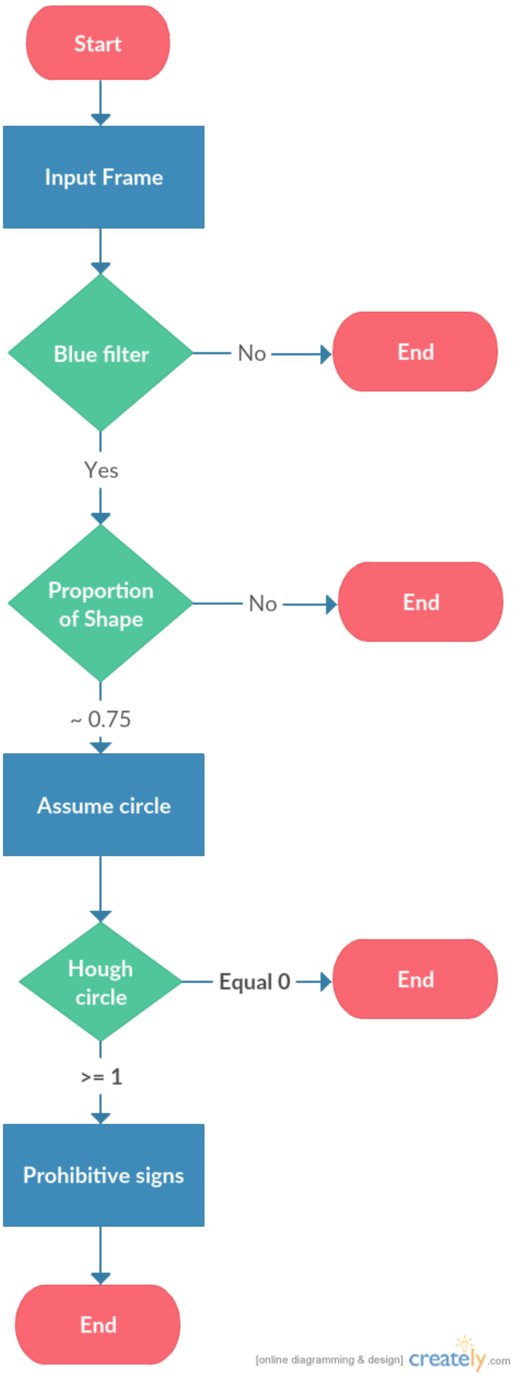

Figure 39. The Mandatory Signs Detection

This is the result after detecting the Mandatory signs. The signs will mark with the bounding rectangle.

	Blue Circle	Red Circle	Figure
Daytime	x	x	Figure 40
	x	x	Figure 41
	x		Figure 42

Table 3. The result of Detection phase in Blue color segmentation

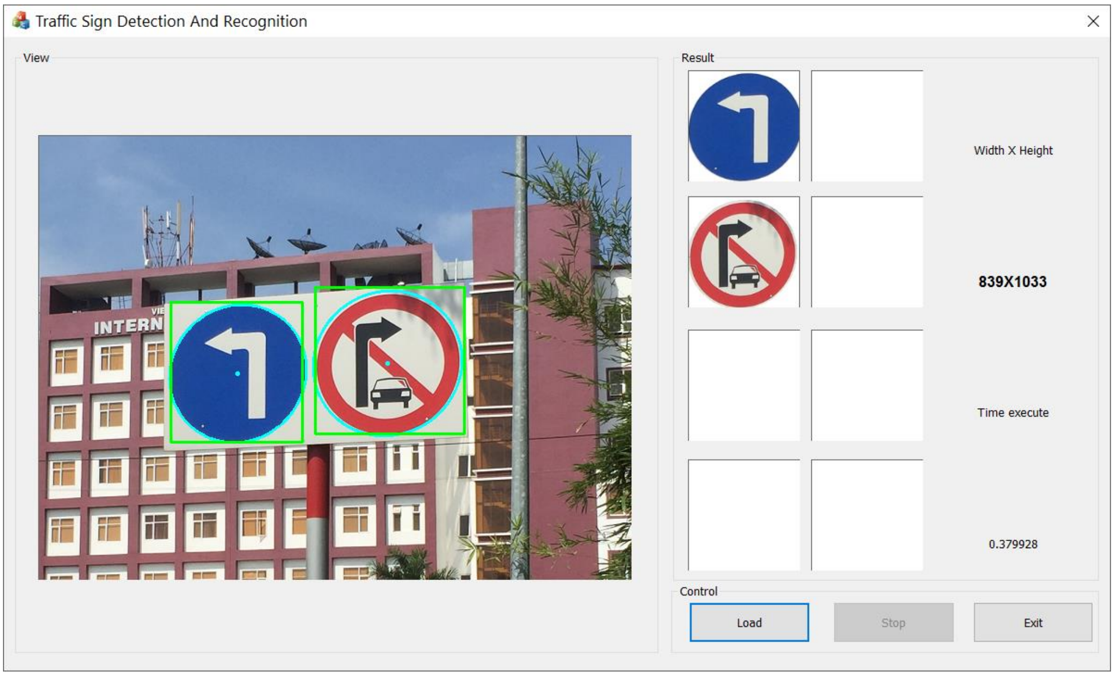

Figure 40. The Mandatory and Prohibitory Signs

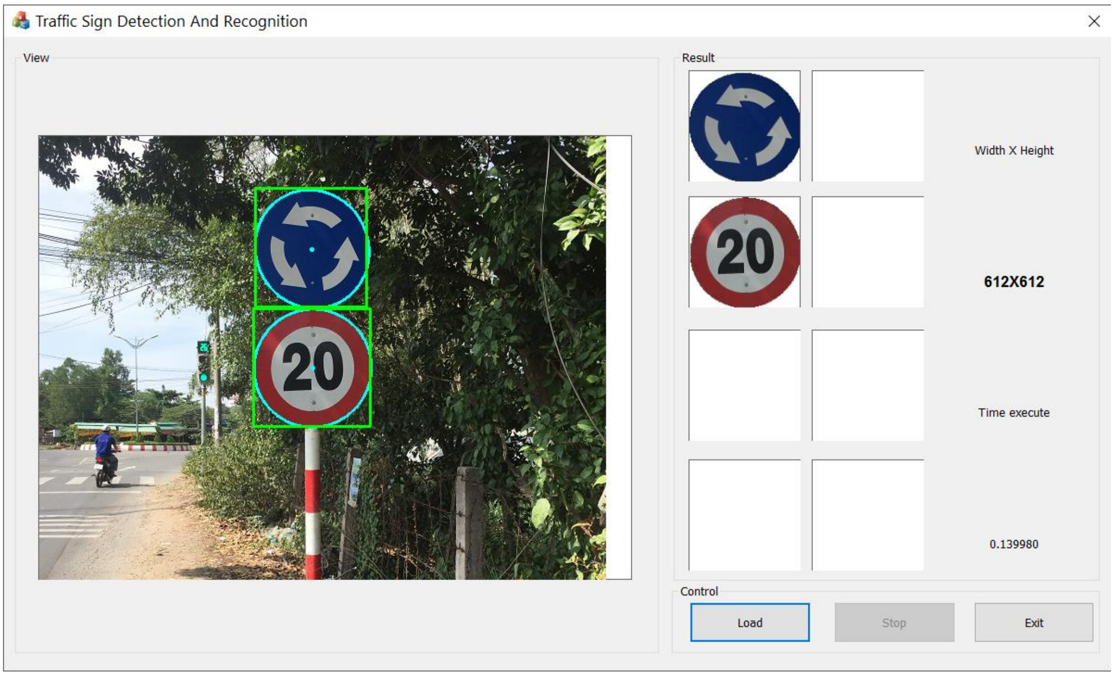

Figure 41. The Mandatory and Prohibitory Signs

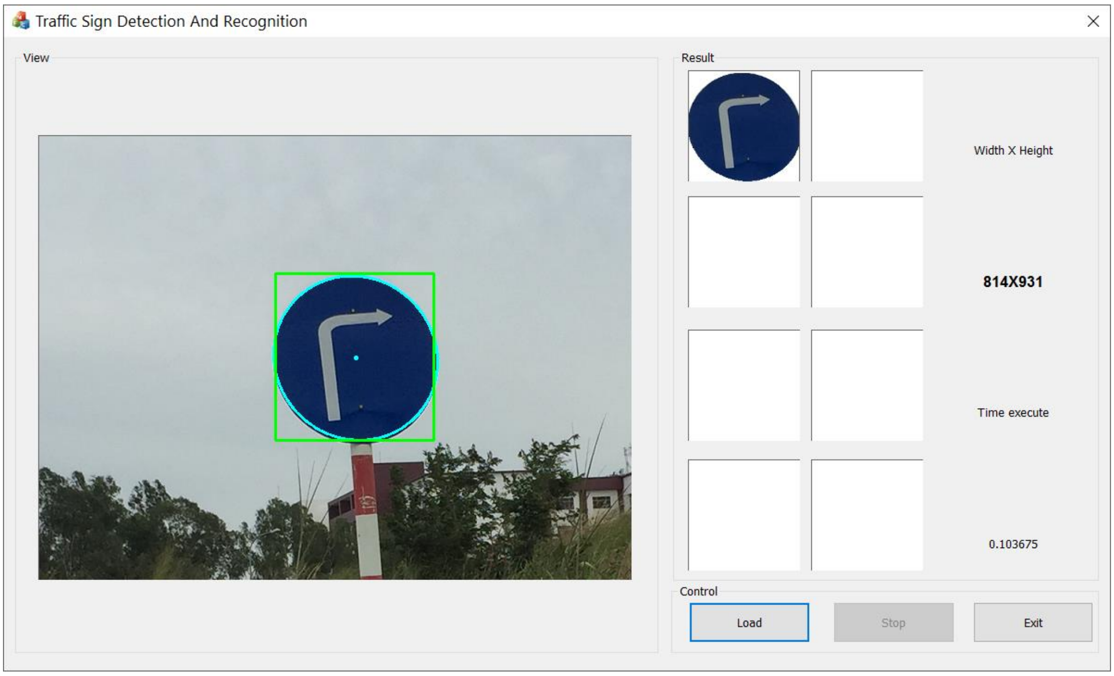

Figure 42. The Mandatory Signs

4.2 Method for Traffic Signs Recognition

The last phase of the Traffic Signs Detection and Recognition is the Traffic Signs Recognition phase (from now will be called by Recognition phase). The annotated traffic signs will be cropped out with the same size as the bounding rectangle in the Detection phase, extract feature and put into the classifier to classify.

In recent years, Support Vector Machine (SVM) method has attracted the attention of research scientists and has gained good recognition results in many recognition problems. SVM classifiers is a new classifier, high performance and can be applied to many different recognition problems. So, SVM classifier is chosen to apply for Recognition phase.

4.2.1 Building the training datasets

- **Step 1:** Building two matrixes to store data

  Matrix 1 will store all of the features of the traffic signs. This matrix has the rows equal the number of traffic signs, the columns equal 2500 corresponds to 2500 pixel in the 50x50 binary image.

  Matrix 2 store the labels (the name of the signs) that indicate the corresponding signs in the matrix 1. This matrix has the same rows of the matrix 1, and the column is one because it needs only one column to store the labels.

- **Step 2:** Set up parameter for training SVM

  svm_type: C_SVC to classify the number of classes equal to or higher than two class.

  kernel_type: LINEAR mean that no mapping is done, linear discrimination (or regression) is done in the original feature space.

  term_crit: this is the standard to terminate the training process. The value can be the tolerance and/or the maximum number of iterations. The chosen value is cvTermCriteria(CV_TERMCRIT_ITER + CV_TERMCRIT_EPS, 1000, 1e-6).

Cvalue: This parameter affects the error rate. If the value is too big, the error rate is smaller, but the training process will be difficult to calculate the weights to satisfy the separation of the traffic signs.

- **Step 3:** Training SVM and save to XML file

  Starting the training process using the above parameter and then save it to XML file to use in the next stage.

4.2.2 Running the training datasets

In this stage, we first classify the candidate objects into 4 categories: prohibitory signs 1 (with blue background), prohibitory signs 2 (with white background), danger signs, and mandatory signs as shown in Table 4.

Attributes	**Prohibitory signs 1**	**Prohibitory signs 2**	**Danger signs**	**Mandatory signs**
Color segmentation	red	red	red	blue
Shape	round shape	round shape	round shape	triangle shape
Background	blue	White	-	-

Table 4. Four categories of Traffic Signs

Next, we convert images of the objects into binary images. To do that, we apply color segmentation and thresholding method for images of the objects as described in Table 5 shows the result of the binary images.

Categories	**Method**
Prohibitory signs 1 & Mandatory signs	Apply blue color segmentation. Blue pixel convert to white one, other convert to black one.
Prohibitory signs 2 & Danger signs	Convert the image to gray scale. Convert the image to binary based on a threshold for pixels inside the object.

Table 5. Methods for converting images of the objects to Binary images

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

- ✓ Resize binary image to 50x50
- ✓ Convert 50x50 matrix to a column vector 2500x1
- ✓ Put the column vector to SVM for classifying the types of traffic signs.

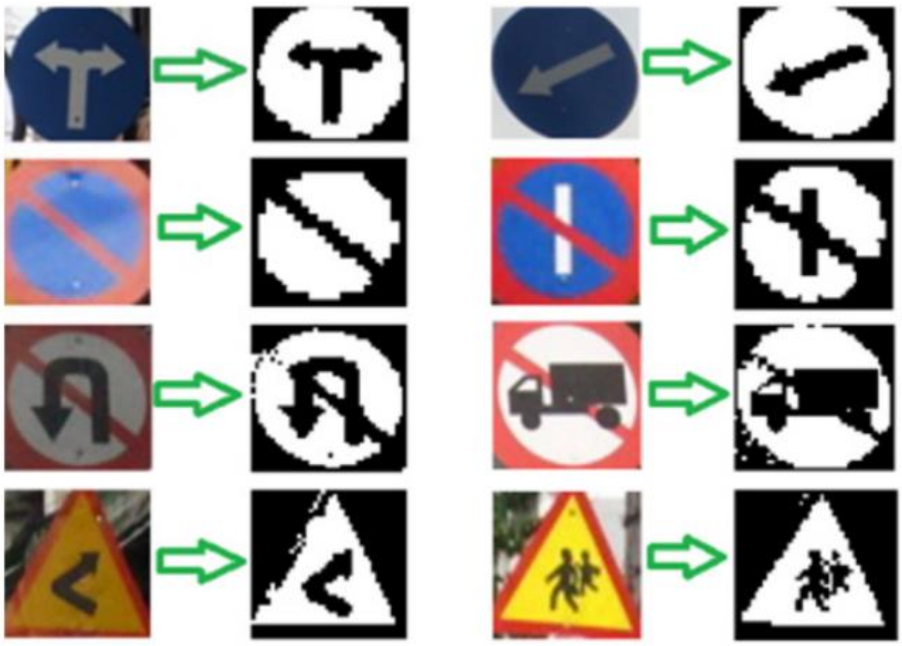

Figure 43. The result of binary image conversion

In order to train the SVM, we first select a training set which contains most popular types of traffic signs in Vietnam including prohibitory signs, danger signs, and mandatory signs. Then, we create feature vectors from samples of binary images of the training set. We make labels for all feature vectors to indicate their types of signs. OpenCV functions from the class CvSVM are used to perform SVM training and classifying.

To recognize the traffic sign, the following steps in the Figure 43 below will be executed:

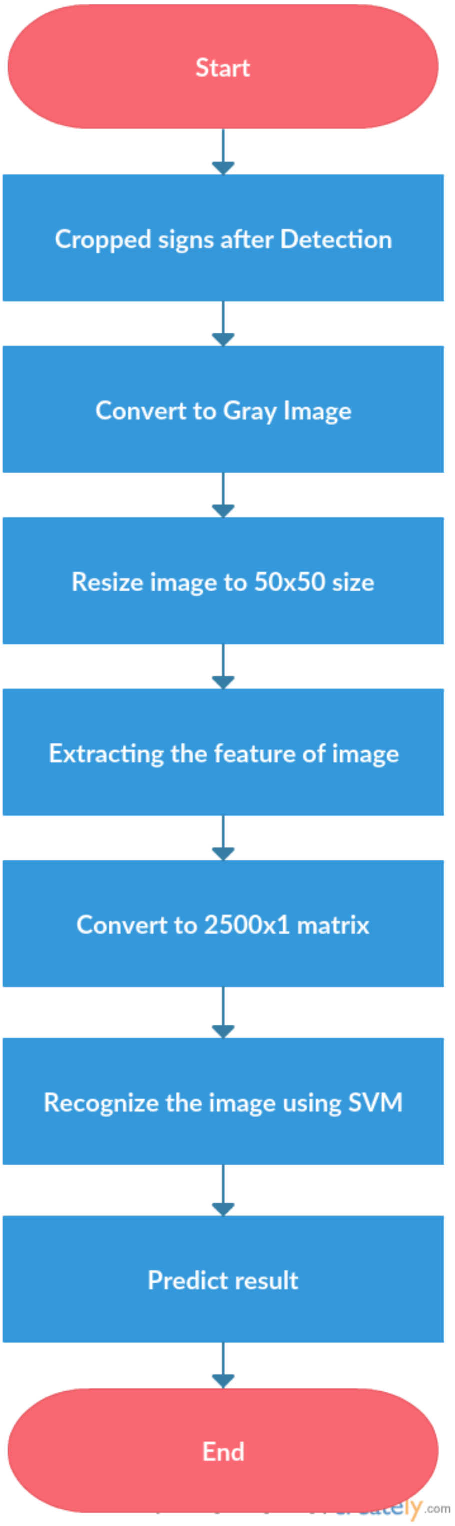

Figure 44. The Traffic Signs Recognition

This is the result of recognizing the traffic signs in the Detection stages. The recognized traffic signs are based on the predict results from SVM and will map with the official traffic signs (which provides by the Ministry of Transport).

Mandatory	Prohibitory	Danger	Figure
x			Figure 44
x			Figure 45
x			Figure 46
	x		Figure 47
	x		Figure 48
		x	Figure 49
		x	Figure 50
		x	Figure 51
		x	Figure 52

Table 6. The result of Recognition phase

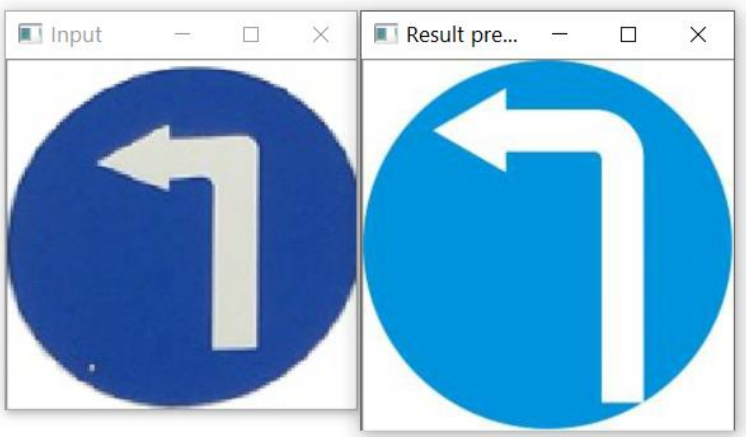

Figure 45. The Recognized Mandatory sign 1

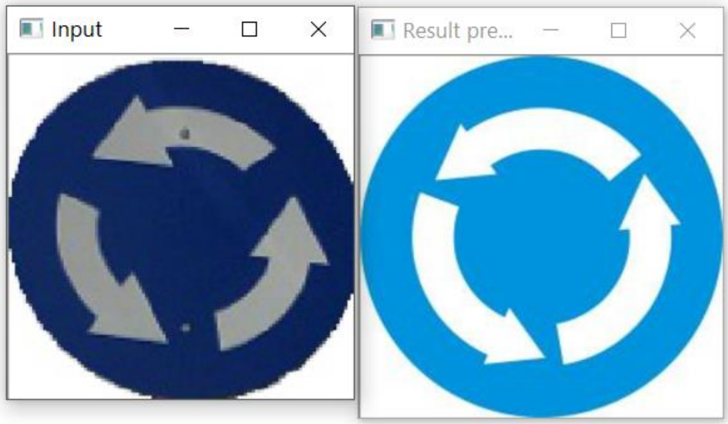

Figure 46. The Recognized Mandatory sign 2

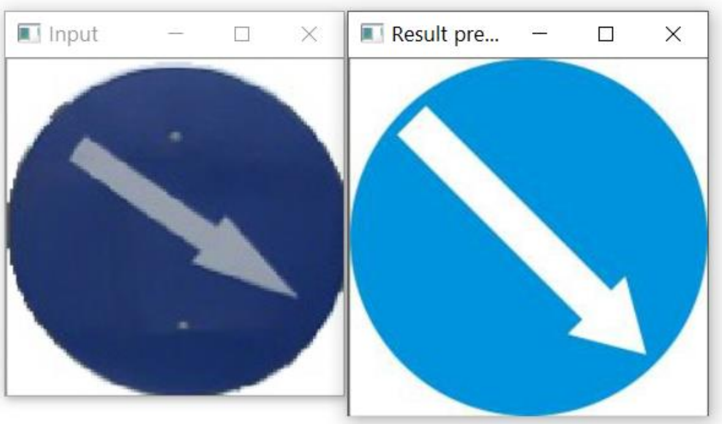

Figure 47. The Recognized Mandatory sign 3

Figure 48. The Recognized Prohibitory sign 1

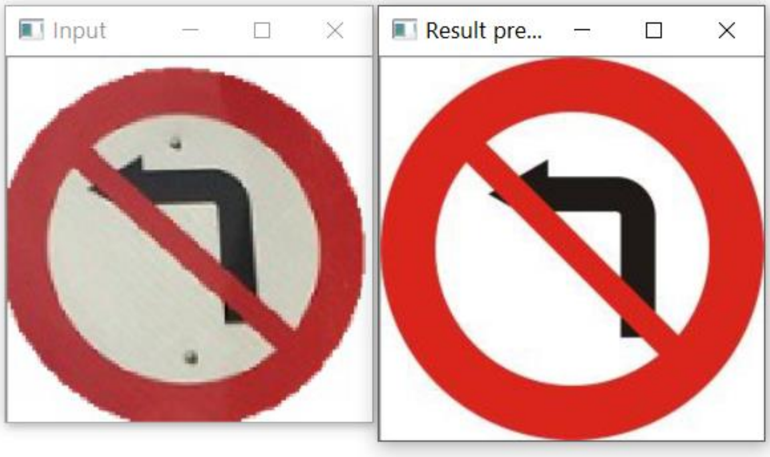

Figure 49. The Recognized Prohibitory sign 2

Figure 50. The Recognized Danger sign 1

Figure 51. The Recognized Danger sign 2

Figure 52. The Recognized Danger sign 3

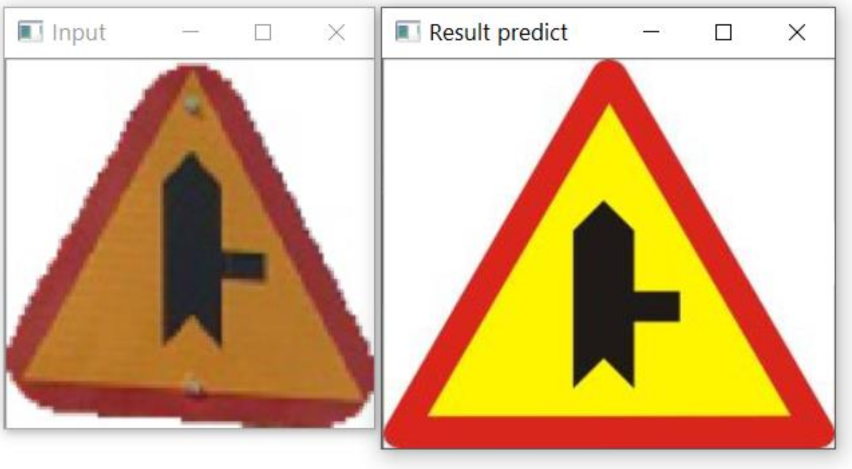

Figure 53. The Recognized Danger sign 4

CHAPTER 5
RESULTS & DISCUSSION

5.1 Results

In this project, the Detection and Recognition of Traffic Signs algorithms involve a combination of computer vision and pattern recognition problems, was able to extract road signs from still images of complex scenes subject to uncontrollable illumination.

After building the program, the program evaluation was testing in different environments on still images and video clips with the size of 640x480 pixels; each clip has a length from 5 to 10 seconds. The test environment assessment is:

- Daytime environment from 7 hours to 17 hours in good weather conditions, no rain, clear signs, flat roads, the car can reach speeds of 35 to 45 km / h. The results showed that the detection rate is 90%, the recognition rate is 90% accurate. This environment is an ideal environment for programs that can detect and identify exactly with the highest rate. (see the Figure above)
- Nighttime environment with the public lighting system and transport's light. With this environment, depending on the quality of the camera, the low light condition, and the light from the cars and advertisement, the light from opposite roadsides. The program is also resulting in 80% of the detection rates, and 79% accurate of the recognition rate. (see the Figure above)
- Daytime environment with mutually occluding, skewed or obscured part (obscured by 10-15% on the total area of sign) signs. The result is 85% of the detection rate and 80% of the recognition rate. (see the Figure above)

5.2 Discussion

As the above results, there are some reasons for the drop in the success rate: the light condition, the camera quality, the physically damaged or faded will also affect the ability to detect and recognize the traffic signs in some cases.

One reason is the failure of algorithms in segmenting yellow. Regardless of the quality of the sign and its age, this failure happens when the image is taken from a certain angle in which a different yellow (hue) can be seen. When the segmentation of yellow fails, recognition of the sign also fails, as it based on combining red-yellow color for sign recognition. This failure in color segmentation is responsible for 30% of all failures in the whole system which is very significant.

Another reason is the Occluded signs. The main cause of this significant drop in performance is that the algorithm is based on shape measures which indicate the similarity of the object under consideration with the basic shapes, i.e. rectangle, ellipse, triangle, and octagon. When the sign is occluded, its shape departs from these basic shapes and the fuzzy system rejects it as it is not close to any of these shapes.

Figure 54. The occluded signs

Therefore more research for the proper method is needed.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The program solves the goal of the study, and it has been testing, and evaluation in many different environments, the number of experimental data sets are large enough to be able to trust the accuracy of the research methods. I also review and evaluate the characteristic element to identify the mistakes of the systems. These results are paving the way for the construction of a system of complete Traffic Signs Detection and Recognition system in Vietnam.

However, because of limited time and knowledge, this research also has many issues to resolved, the most representative is the recognition rate is not high. Moreover, the real traffic signs in Vietnam is also an enormous difficulty, affecting the process of detection and recognition of the algorithms such as: mutually occluding, skewed or obscured part signs, the light and environment condition.

6.2 Future work

In subsequent research, the system will overcome the limitations on the collection of datasets using special cameras for better image quality, minimize mistaken of identity between signs. The expansion of the database traffic signs is critical and necessary in the next research direction.

In the future, overcoming the limitations of the study, the program may expand the scope of research to the traffic signs in Vietnam and then can be combined with other systems such as pedestrian recognition, automatic braking system to the driver can be safer. This research topic suitable for the development of automotive systems automatically.

REFERENCES

[1] C. Fang, S. Chen, and C. Fuh,, "Road-sign detection and tracking," *IEEE Trans. On Vehicular Technology,* vol. 52, pp. 1329-1341, 2003.

[2] H. Fleyeh, "A Novel Fuzzy Approach for Shape Determination of Traffic Signs," in *The Second Indian International Conference on Artificial Intelligence (IICAI-05),* Pune, India, 20-22 December, 2005.

[3] P. Parodi and G. Picciol, "A feature-based recognition scheme for traffic scenes," in *The Intelligent Vehicles '95 Symposium*, Detroit, USA, 1995.

[4] J. Plane, Traffic Engineering Handbook, Upper Saddle River, New Jersey: Englewood Cliffs, N.J. : Prentice-Hall, 1992.

[5] G. Jiang and T. Choi, "Robust detection of landmarks in color image based on fuzzy set theory," in *The Fourth Inter. Conf. on Signal Processing*, Beijing, China, 1998.

[6] S. Vitabile, A. Gentile, and F. Sorbello, "A neural network based automatic road sign recognizer," in *The 2002 Inter. Joint Conf. on Neural Networks*, Honolulu, HI, USA, 2002.

[7] J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition," in *The 2000 IEEE Intelligent Transportation Systems*, USA, 2000.

[8] S. Vitabile, A. Gentile, G. Dammone, and F. Sorbello, "Multi-layer perceptron mapping on a SIMD architecture," in *The 2002 IEEE Signal Processing*, 2002.

[9] S. Vitabile, G. Pollaccia, G. Pilato, and F. Sorbello, "Road sign Recognition using a dynamic pixel aggregation technique in the HSV color space," in *The 11th Inter.Conf. Image Analysis and Processing*, Palermo, Italy, 2001.

[10] P. Paclik and J. Novovicova, "Road sign classification without color information," in *The Sixth Annual Conf. of the Advanced School for Computing and Imaging*, Belgium, 2000.

[11] P. Paclik, J. Novovicova, P. Pudil, and P. Somol, "Road sign classification using Laplace kernel classifier," *Pattern Recognition Letters*, vol. 21, pp. 1165-1173, 2000.

[12] E. Perez and B. Javidi, "Composite filter bank for road sign recognition," in *The 13th Annual Meeting IEEE Lasers and Electro-Optics Society*, Rio Grande, Puerto Rico, 2000.

[13] e. a. A. Broggi, "Real Time Road Signs Recognition," *Intelligent Vehicles Symposium*, vol. 2007 IEEE, pp. 981- 986, 2007.

[14] H. Yea-Shua and L.Yun-Shin, "Detection and recognition of speed limit signs," *Computer Symposium (ICS)*, vol. 2010 International, p. 107 – 112, 2010.

[15] e. a. C. Hsin-Han, "Road speed sign recognition using edge-voting principle and learning vector quantization network," *Computer Symposium (ICS)*, vol. 2010 International, pp. 246-251, 2010.

[16] C. Y. Fang, C. S. Fuh, S. W. Chen, and P. S. Yen, "A Road Sign Recognition System Based on Dynamic Visual Model," *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I-750-I-755, June, 2003.

[17] W. Shadeed, D. Abu-Al-Nadi, and M. Mismar, "Road traffic sign detection in color images," in *The 10th IEEE Inter. Conf. on Electronics, Circuits and Systems (ICECS 2003),* Sharjah, United Arab Emirates, 2003.

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

[19] M. Shneier, "Road Sign Detection and Recognition," in *The IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, June 2005.

[20] M. Benallal and J. Meunier, "Real-time color segmentation of road signs," *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference*, vol. vol.3, p. 1823–1826, May 2003.

[21] T. B. Moeslund, Image and Video Processing 2nd Edition, Cambridge, Massachusetts: Academic Press, June 21, 2005.

[22] I. The MathWorks, "http://www.mathworks.com/," MathLabs language, [Online]. Available: http://www.mathworks.com/help/images/ref/imfindcircles.html.

[23] Thanh Bui-Minh, Ovidiu Ghita, Paul F. Whelan and Trang Hoang, "A Robust Algorithm for Detection and Classification of Traffic Signs in Video Data," in *The 2012 International Conference on Control, Automation and Information Sciences (ICCAIS),* Ho Chi Minh City, 2012.

[24] David Soendoro, Iping Supriana, "Traffic sign recognition with Color-based Method, shape-arc estimation and SVM," in *The 2011 International Conference on Electrical Engineering and Informatics*, Bandung, Indonesia, 17-19 July 2011.

[25] Lutz Priese, Jens Klieber, Raimund Lakmann, Volker Rehrmann, Rainer Schian, "New results on traffic sign recognition," in *Proceedings of the Intelligent Vehicles '94 Symposium*, 1994.

[26] Piccioli, G., De Micheli, E., Parodi, P., Campani, M., "Robust method for road sign detection and recognition," *Image Vis. Comput. 14(3)* , vol. 3, p. 209–223, 1996.

[27] H. Ohara, I. Nishikawa, S. Miki, and N. Yabuki, "Detection and recognition of road signs using simple layered neural network," in *The 9th Inter. Conf. Neural Information Processing,* Singapore, 2002.

[28] S. Lafuente-Arroyo, P. Gil-Jim´enez, R. Maldonado-Basc´on, F. L´opez-Ferreras, and S. Maldonado-Basc´on, "Traffic sign shape classification evaluation I: SVM using distance to borders," in *The IEEE Intelligent Vehicles Symposium*, Las Vegas, USA, 2005.

[29] Qingzhou Mao, Yingying Yu, Qin Zou, "Novel Approach to Traffic Sign Recognition Based on SIFT-Matching," in *The 3rd International Conference on Machine Vision (ICMV 2010),* Hong Kong, 2010.

[30] I. Wikimedia Foundation, "Wikipedia," Wikimedia Foundation Inc. A Non-profit Organization, 2001. [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine.

[31] Alexander Mordvintsev & Abid K. Revision, "OpenCV-Python Tutorials," 2013. [Online]. Available: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_svm/py_svm_basics/py_svm_basics.html.