Traffic Sign Segmentation

BY

Dai, Cheng Xiao
Lee Jian Shen
Tan Lin
Paun Tek You

A PROPOSAL
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfilment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONS)
Faculty of Information and Communication Technology
(Kampar Campus)

July 2023

TABLE OF CONTENTS

ABSTRACT

In recent years, Artificial Intelligence (AI) has gained significant importance across various industries and applications. Its diverse capabilities have transformed the way we interact with technology and led to breakthroughs in fields such as computer vision (CV), which focuses on enabling machines to interpret and comprehend visual information from their surroundings. CV has revolutionized numerous sectors, including transportation, where it plays a vital role in enhancing road safety and efficiency. A subspecific area within CV is traffic sign detection and segmentation which aims to automatically identify and delineate traffic signs from complex real-world scenes.

In this paper, a traffic sign colour based which is a traditional image processing method will be proposed that intentionally develop for aiding the self-driving automobile to segment out the traffic sign on the road as there are a lot of kinds of the traffic sign board representing different meanings and warnings to the road user with different shapes and colours so it might be a challenge for the automobile sensor and camera to segment out the images.

LIST OF FIGURES

List of tables

Formatted: Normal

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CV	Computer Vision
OpenCV	Open Source Computer Vision Library
CNN	Convolution Neural Network
TSR	Traffic Sign Recognition
ADAS	Advanced Driver Assistance Systems
FPN	Feature Pyramid Network
RPN	Region Proposal Network
GTSDB	German Traffic Sign Detection Benchmark
TRSD	Traffic Sign Recognition Database
SVM	Support Vector Machine

Chapter 1
Introduction

1.1 Problem Statement and Motivation

In recent years, technology is evolving rapidly and how we live, work, communicate, and engage with the world around us has substantially changed as a result of technological advancements in many different industries.

One of the dominant topics of technology is autonomous driving vehicles also known as auto-driving vehicles or self-driving cars have become a significant trend in the automobile industry. For developing an autonomous driving vehicle, technologies such as image segmentation and image pre-processing are playing an important role and are popularly used in extracting meaningful information from the images captured. For instance, image segmentation aids the auto-driving vehicles in segmenting out the traffic sign from the board on the road in the possible shortest time. However, there might exist certain difficulties when processing the traffic sign segmentation due to the factors such as shape variabilities, position and perspectives, weather conditions and some specific road scenarios. The difficulties stated above might challenge the traffic sign segmentation process as the image captured might not be clear enough or of low quality.

Accordingly, to tackle the difficulties, a colour-based traffic sign segmentation method will be proposed in this proposal paper. Image pre-processing such as Gaussian blurring, scaling and brightness correcting will be performed on the captured images to enhance the image quality before performing the image segmentation tasks. The expected outcome is to improve the traffic sign segmentation method so to enhance the experience of self-driving cars.

1.2 Project Scope, Objectives

1.2.1. Project Scope

The aim of this project is to utilize traditional image processing techniques to achieve colour-based traffic sign segmentation. In this project, we will develop an algorithm using the Python programming language and the Open Source Computer Vision Library [OpenCV] to accurately segment traffic sign regions from images in the Traffic Sign Recognition Database [TRSD] dataset.

In the implementation of the project, we first preprocess the input images, including image resizing to uniform size, noise reduction, and image enhancement to improve image quality and segment ability. Secondly, the traffic sign region is separated from the background by using a color-based segmentation technique. According to the color characteristics of traffic signs, we will select the appropriate color threshold for segmentation. Next, we will apply the edge detection technique to detect the edges and contours of the segmented traffic sign region, which will help define the shape of the traffic sign and accurately locate it. The edge information detected in the previous step is subsequently used to form the outline of the traffic sign area, which will be used to calculate the bounding box to facilitate the creation of a rectangular box surrounding the traffic sign area. Finally, according to the calculated bounding box, we will extract the traffic sign area to obtain the segmented traffic sign.

As mentioned above, this project will mainly focus on segmenting traffic signs based on colour using traditional image processing methods that can be applied to autonomous driving in the future.

1.2.2. Project Objectives

The goal of this project is to use traditional image processing methods to achieve colour-based traffic sign segmentation, in order to help advanced Driver assistance systems [ADAS] improve traffic sign recognition [TSR] function, reduce the burden of drivers in the driving process, improve driving safety, and lay the foundation for the development of future autonomous driving technology. The objectives of this project are:

Colour-based traffic sign segmentation using traditional image processing methods.

Enabling it to accurately segment traffic sign images under different conditions and scenarios.

1.3 Impact Significance and Contributions

Our proposed model for traffic sign segmentation would be greatly beneficial for the accuracy and efficiency of traffic sign detection and recognition systems in various environmental conditions. By detecting and segmenting traffic signs, the system can proactively identify potential hazards and respond appropriately to various traffic regulations and warnings. This contributes to improved overall road safety as the system can make informed decisions based on the recognized traffic signs.

The project's contributions extend beyond autonomous vehicles, as traffic sign segmentation has practical applications in driver assistance systems for conventional vehicles with ADAS. In addition, the generation of large datasets with annotated traffic signs through segmentation can serve as valuable data augmentation for training and fine-tuning deep learning models. These enhanced models are more effective in handling real-world scenarios, contributing to the ongoing advancement of computer vision and autonomous systems technology. In conclusion, the impact and contributions of traffic sign segmentation are instrumental in promoting road safety, enhancing navigation, and revolutionizing the transportation industry.

1.4 Background Information

Traffic sign segmentation is crucial for computer vision and image processing, particularly in the context of advanced driver assistance systems (ADAS) and autonomous vehicles. This process involves automatically identifying and segmenting traffic signs from images or video frames. The segmentation allows for recognizing and classifying traffic signs into various categories, such as speed limits, stop signs, and yield signs [1]. In the past, traffic sign segmentation faced challenges due to variations in shape, color, size, orientation, complex backgrounds, lighting conditions, and computational costs [2]. However, recent advances in computer vision and machine learning have led to more effective and efficient segmentation methods [3,4]. The success of traffic sign segmentation directly impacts the safety and efficiency of autonomous vehicles, as it enables tasks like adjusting vehicle speed based on speed limit signs, detecting stop signs, and recognizing various other types of signs [2].

This project is an attempt to build a model that can accurately segment traffic signs from images using traditional image processing techniques. In the following sections, we will review some of the existing methods for traffic sign segmentation and analyse their advantages and limitations. We will also discuss the dataset that we will use for our project, which is the Traffic Sign Recognition Database (TRSD), and its characteristics and challenges. Finally, we will present our proposed method for traffic sign segmentation based on colour using traditional image processing techniques and explain its rationale and motivation.

Chapter 2
Literature Review

2.1 Traditional Method

Done by: Paun Tek You

Image processing is an important topic to be discovered, previous studies have been dedicated to exploring computer vision methods for segmenting and identifying traffic signs in images or videos. These methods serve as the foundation of early approaches used in various applications, including advanced driver assistance systems (ADAS) [1]. The following section is an overview of the key “Traditional Methods” employed for traffic sign segmentation will be presented.

2.1.1 Smart data driven traffic sign detection method based on adaptive colour threshold and shape symmetry.

In this paper, an adaptive colour threshold segmentation method has been proposed by the authors. Firstly, the dataset which is referring to the traffic sign images initially perform the Red-Blue normalization which is a pre-processing step to convert the images into Red-Blue Gray images [7]. The purpose of doing this is to enhance the red and blue colour channel as the traffic signs having a large scale of red and blue colours.

However, the paper stated that the foreground sign is still unable to be highlighted due to the surroundings factor such as the high contrast and the brightness. Therefore, the proposed method called approximate maximum-minimum normalization play a role at this stage to overcome the problem[7]. A cumulative distribution function (Figure 2.1.1.4) is interpreted from the image histogram (Figure. 2.1.1.1)

$$F\,(m) = \sum_{k=min}^{m} p(k)$$

Figure. 2.1.1.1 Cumulative distribution function of a Red–Blue image [7]

According to the formula that shown in figure 2.1.1.2 and further analysis by author [7], a solution to find out the adaptive threshold has been interpreted: Given β = [0,1]

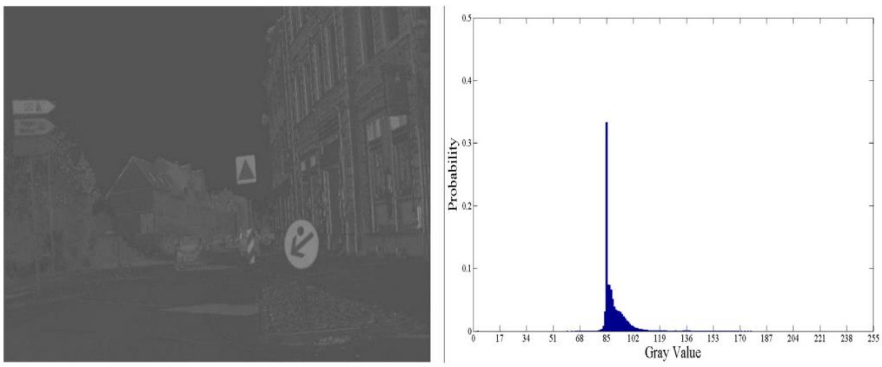
$$\text{Finding } m = m_1 \text{ to satisfy} \quad \begin{cases} F(m_1 - 1) < \beta & \text{and} \\ F(m_1) \geq \beta \end{cases}$$

Figure 2.1.1.2 Adaptive threshold selection [7]

Lastly, the adaptive threshold (m1) will be used to perform the normalization process which follow the setting below (Figure 2.1.1.3):

$$m_1: \quad RB'(x) = \begin{cases} 0, & RB(x) < m_1 \\ 255, & RB(x) = m_1 \\ \frac{RB(x) - m_1}{max(RB) - m_1} * 255, & RB(x) > m_1 \end{cases}$$

Figure 2.1.1.3 Approximate maximum–minimum normalization equation. [7]



(a) Red-Blue image

(b) Probability density function of Red-Blue image

Figure. 2.1.1.4 Probability histogram in Literature Review 2.1.1[7]

As a result, the adaptive threshold normalization has successfully minimized the effect of low-quality images due to brightness and contrast.[7] Traffic sign successfully to be segmented and the results are shown on the figure (Figure 2.1.1.5) below.

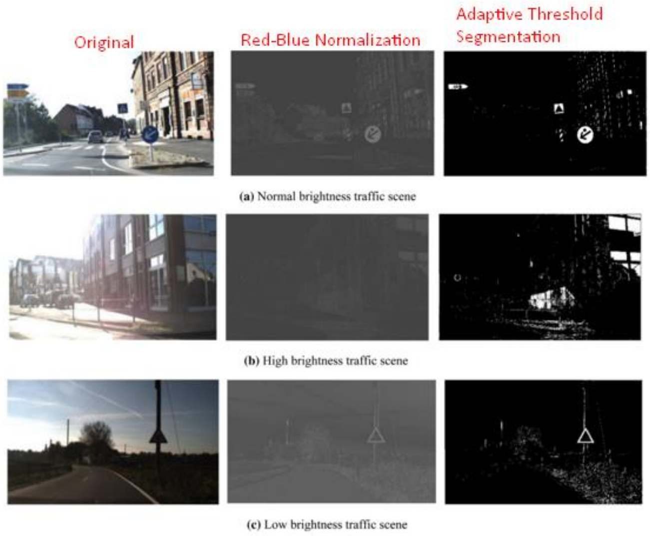


Figure 2.1.1.5 Image processing results [7]

Lastly, figure (Figure 2.1.1.6) below showing system flow of the method proposed in the reviewed paper however only first three steps are being reviewed in the previous paragraph and useful for this proposal as they are related to the traffic sign segmentation which are the Data Input, Red-Blue Normalization and Approximate Max-Min Normalization step. To summarize up, this colour adaptive threshold segmentation with the max-min normalization is useful in aiding the segmentation in different environment. However, the proposed method might be sensitive to the image quality as the method is based on the image histogram analysis. Therefore, image pre-processing such as image denoising might needed to overcome the weakness.

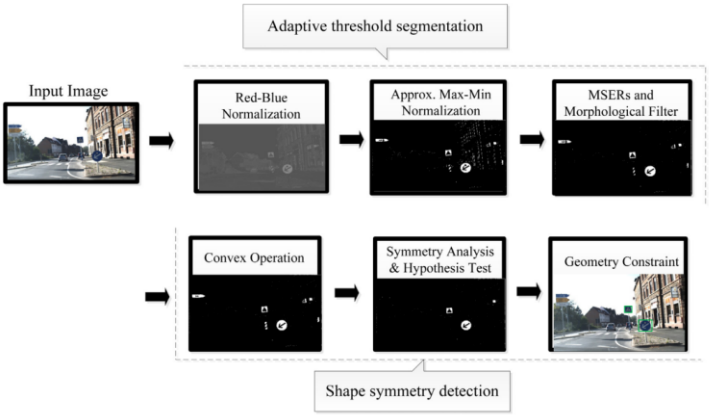


Figure 2.1.1.6 Flow chart of the traffic sign detection process [7]

2.1.2 Image Segmentation and Shape Analysis for Road-Sign Detection

Colour featuring clustering has been proposed in this paper [8] to aid the traffic sign segmentation. During the initial step of the segmentation, the Gabor Filter is used as a feature extraction to extract the texture feature and the from the CIE LAB colour space and the local energy from a* which represents the colours belongs to red-green axis and b* represent the colours falls on blue-yellow axis.[8] The equation below is the local energy that derived from a*:

$$\mathrm{LE}^{a^*}(x,y) = \left(a^*(x,y) ** G_\theta^{o}(x,y)\right)^2 + \left(a^*(x,y) ** G_\theta^{e}(x,y)\right)^2 \qquad (1)$$

Figure 2.1.2.1 Derived formula for the a∗ component [8]

A Gabor Filter has been optimized after several trial error of clustering at an angle with different images:

$$G_\theta(x,y) = e^{-\frac{x^2+y^2}{18}} \left(\cos\left(0.2\pi(x\cos\theta + y\sin\theta)\right) + j\sin\left(0.2\pi(x\cos\theta + y\sin\theta)\right)\right).$$

Figure 2.1.2.2 Optimized Gabor filter [8]

K-means clustering

A 2-D feature space has been obtained after the feature extraction. In addition, K-means clustering is then being performed to gather the similar points and the value of K is tested and selected for a better segmentation. In addition, pixel-clustering is then used to create clustered image.[8]

Post-Processing on Clustered Image

In order to obtain a more unified and good quality segmented image, some approaches have been done by authors. For instance, the image is being spatial smooth by a 3x3 mode until convergence condition is good enough.[8] Verification process of blobs is then performed on the segmented image and thresholds are set to retain the blobs that fulfil the requirements (aspect ratio) [8]. Finally, the Region of Interest is successfully obtained in Figure 2.1.2.3 below

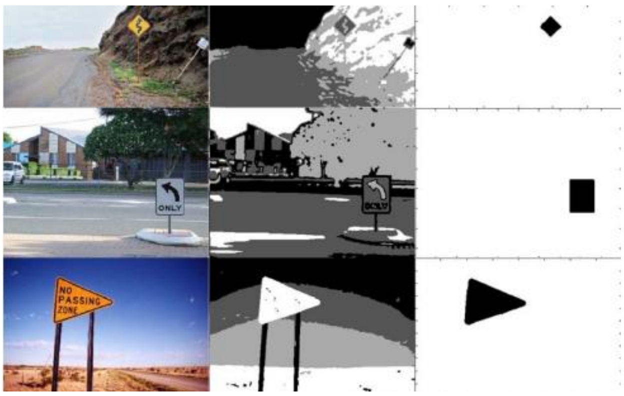


Figure 2.1.2.3 Segmentation results [8]

In the figure (Figure 2.1.2.3) above, the respective column is the original traffic sign images, segmented image and final result of the traffic sign segmentation after performing the spatial smooth and blobs unified process.

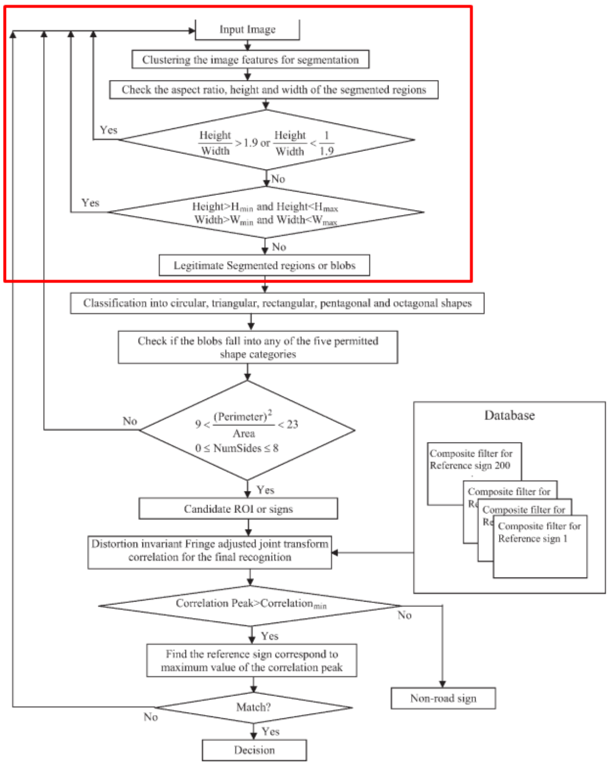


Figure 2.1.2.4 Flow diagram of the proposed algorithm [8]

Figure (Figure 2.1.2.4) above shows the flows of the segmentation process proposed in the research paper which is discussed in this section. However, only the first six steps are interesting to discuss in this section as they are related to traffic sign segmentation procedures.

2.2 Machine Learning Method

Done by: Tan Lin

Machine learning is a branch of AI that enables computers to learn from data and perform tasks such as classification, regression, clustering, and anomaly detection. Machine learning techniques have been widely applied in various fields, including computer vision and image processing. In this section, we will review some machine learning methods for traffic sign segmentation.

2.2.1 Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition

Hilario et al. [11] conduct a comparative study on segmentation methods for traffic sign recognition. They explore color-based techniques, edge detection methods like Laplacian and Canny, as well as SVMs. In this section, we will focus on the efficacy of SVMs for color segmentation in traffic sign recognition.

SVMs are known for their adaptability to diverse scenarios and their ability to learn from sample data, making them suitable for handling color classification tasks. According to the authors, the advantages of SVMs in image segmentation lie in their capacity to provide a unique and optimal solution, showcasing robustness and generalization capabilities while simultaneously reducing the need for a large number of parameters and thresholds.

However, the authors also acknowledge certain limitations associated with SVMs. Firstly, SVMs can be computationally slow and necessitate extensive training data and the use of kernel functions. Secondly, SVMs may require improvements for effectively handling achromatic colors, such as white or gray, which can pose challenges during segmentation.

The authors propose several SVM-based methods for image segmentation, specifically tailored to traffic sign recognition. They consider different types of SVMs, such as linear, polynomial, radial basis function (RBF), and sigmoid kernels, while also exploring various ways of utilizing them, including one-class SVM, multiclass SVM, and hierarchical SVM approaches.

For their experimental evaluations, the authors implement the proposed methods using Matlab and LIBSVM, a library specifically designed for SVM algorithms. They employ diverse datasets of traffic sign images [Figure 2.2.1.1] to test the effectiveness of their methods in segmenting different types of signs.

Figure 2.2.1.1 Some frames from each testing set [11]

Multiple sequences of traffic sign images taken from different routes and under various lighting conditions are encompassed. Each sequence comprises thousands of images. To assess the most challenging scenarios, several sets are extracted, each containing frames presenting particular segmentation difficulties. These problematic situations encompass diverse challenges, including low illumination, rainy conditions, an array of signs, similar background colors, and occlusions. Representative frames from these sets are displayed in Figure 2.2.1.1.

For the analysis present in this paper, a total of 313 images are meticulously selected from thousands of 800 × 600 pixel images to serve as a representative sample of the primary segmentation problems encountered.

The evaluation is conducted to assess the overall performance of the recognition system by examining correctly recognized signs with various segmentation methods while keeping the detection and recognition components unchanged. The evaluated parameters include: 1. the number of correctly recognized signs (normalized score), 2. global rate of correct recognition, 3. number of lost signs (not recognized), 4. frequency of achieving the maximum score for each method, 5. false recognition rate, and 6. speed (execution time per frame).

These parameters are computed using an automatic tool in a Linux environment with a 2.6.27 kernel, comparing the results with known ground truth data for all sequences. The evaluation results for achromatic methods, exemplified in Figure 2.2.1.2, demonstrate the effectiveness of specific techniques for individual signs in the sequence.

Sign	RGBN	Ohta	SI [28]	CAD [14]	RGB
1	0.72	0.44	***0.78***	0.00	0.67
4	***0.67***	***0.67***	0.44	0.44	0.00
5	***0.14***	***0.14***	***0.14***	***0.14***	***0.14***
6	***1.00***	0.93	0.93	0.13	0.60
8	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00
10	***0.88***	0.75	0.75	0.62	0.75
11	0.62	0.62	0.50	0.38	***0.75***
12	***0.62***	***0.62***	***0.62***	0.25	***0.62***
13	***0.45***	***0.45***	***0.45***	0.36	0.36
16	0.80	***0.90***	***0.90***	0.80	***0.90***
17	***1.00***	***1.00***	***1.00***	***1.00***	0.57
18	0.36	***0.71***	0.57	0.43	0.36
20	0.75	0.62	0.62	0.75	***1.00***
21	***1.00***	0.62	0.75	0.88	0.62
22	***0.88***	0.81	0.81	0.69	***0.88***
25	0.00	0.00	0.00	0.00	0.00
26	0.82	***0.89***	0.82	0.61	0.41
28	***0.90***	***0.90***	***0.90***	***0.90***	***0.90***
29	***0.68***	***0.68***	***0.68***	***0.68***	***0.68***

Figure 2.2.1.2 Results of the different achromatic methods studied [11]

Among the methods using SVM, SVMC (SVM Color Segmentation) achieves the second-highest total score (28.5) and global rate of correct recognition (97.6%), yet it exhibits a relatively high number of lost signs (4) and a false recognition rate of 1.2%. However, SVMC's drawback lies in its slow speed (0.3 s/frame) and the requirement of training data and kernel functions. HSET (Hue and Saturation Color Enhancement Thresholding), another method employing SVM, obtains a lower total score (26.8) and global rate of correct recognition (94.9%), but displays fewer lost signs (2) and a lower false recognition rate of 0.6%. It also achieves a faster speed (0.8 s/frame) due to its use of lookup tables, but it remains slower compared to other methods. Despite the good results obtained with

SVM in color segmentation, the authors emphasize the need for improvements concerning achromatic colors and speed enhancement.

2.2.2 Real-Time Traffic Sign Recognition System based on Colour Image Segmentation

This paper [12] introduces a novel traffic sign recognition system that leverages color image segmentation and joint transform correlation. The system comprises three fundamental phases: frame selection, segmentation and detection, and recognition. While the recognition phase has been extensively studied in the literature, this section will delve into the segmentation and detection phases to explore the techniques employed in extracting the region of interest (ROI) from the selected frame.

In the frame selection phase, the system utilizes a video stream captured by a web camera, operating at a frame rate of 30 frames per second (fps) and a frame resolution of 320x240 pixels. The video stream is converted into 'avi' format for further processing. To improve the efficiency of the system, each frame is downsampled to a size of 160x160 pixels after carefully selecting frames that are likely to contain candidate traffic signs.

To evaluate the system performance, a substantial number of images are tested, sourced from publicly available video streams on platforms like Google and YouTube. Representative sample images are shown in Figure 2.2.2.1, providing an overview of the dataset used for evaluation.

Figure 2.2.2.1 Frame from video string [12]

The segmentation, detection, and recognition processes in the proposed method are illustrated in a series of figures. Figure 2.2.2.2 displays the general architecture for real-time traffic sign recognition. Figure 2.2.2.1 showcases the input frame, carefully selected as the most suitable frame for further processing. To facilitate color channel separation, Figure 2.2.2.3 exhibits the frame after it is divided into its three RGB (Red, Green, and Blue) color channels. This step is crucial for subsequent processing.

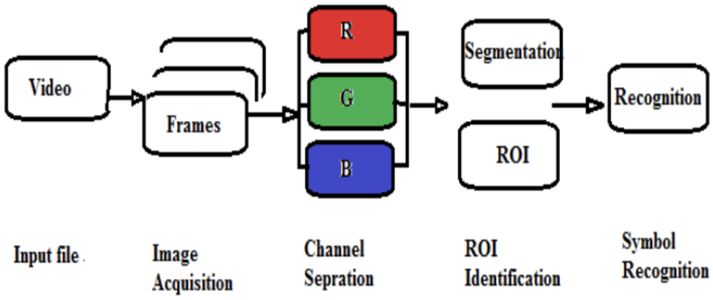


Figure 2.2.2.2 General architecture for Real-Time traffic sign recognition [12]

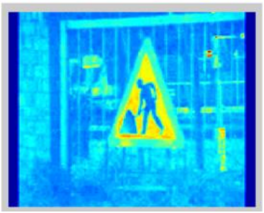


Figure 2.2.2.3 RGB Channel Separation Image [12]

Figure 2.2.2.4 demonstrates the importance of segmentation, as it reveals the segmented part of the color image. This segmented part significantly contributes to the overall performance of the system. In Figure 2.2.2.5, the ROI is shown. This region is extracted from the given frame and segmented based on an edge-based approach. The ROI contains critical information for traffic sign recognition.

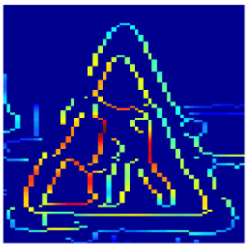


Figure 2.2.2.4 Segmented Image by Using Edge Base [12]

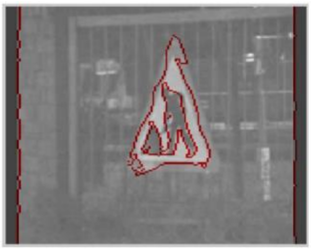


Figure 2.2.2.5 Region of Interest [12]

Subsequently, the system employs the joint transform correlator to match the region of interest with images in the database, enabling traffic sign recognition. Figure 2.2.2.6 presents the output of the recognition process, indicating the successfully recognized traffic sign.

Figure 2.2.2.6 Output of Recognition Phase [12]

The final step involves generating a graph of the Joint Transform Correlator results, and it provides insights into the effectiveness of the traffic sign recognition process.

2.3 Hybrid of machine learning and traditional Methods

Done by: Lee Jian Shen

2.3.1 A novel traffic sign detection method via color segmentation and robust shape matching

In this paper [9], the author introduced a new traffic sign detection method by through the use of color invariance-based image segmentation and shape matching based on Pyramid Histogram Oriented Gradient (PHOG) features.

This approach first extracts the color invariance of the target image and segments the image into different regions to obtain candidate regions of interest (ROIs). Then, PHOG is employed to represent the shape properties of ROIs, and support vector machines are used to recognize traffic signs. The article proposes a new traffic sign detection method that can robustly detect traffic signs under various weather, shadow, occlusion, and complex background conditions .

The article also details how to detect traffic signs using image segmentation based on color invariance and shape matching based on PHOG features. First, candidate regions of interest are obtained by extracting color invariance in a Gaussian color model, and then segmenting the image into different regions. Figure 2.3.1.1 shows the equation of linear translation from RGB to Gaussian. To improve the discriminative ability of PHOG, a method is proposed to introduce color edges to enhance object contours while suppressing noise.

$$\begin{bmatrix} E \\ E_{\lambda} \\ E_{\lambda\lambda} \end{bmatrix} = \left\langle \begin{matrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{matrix} \right\rangle \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

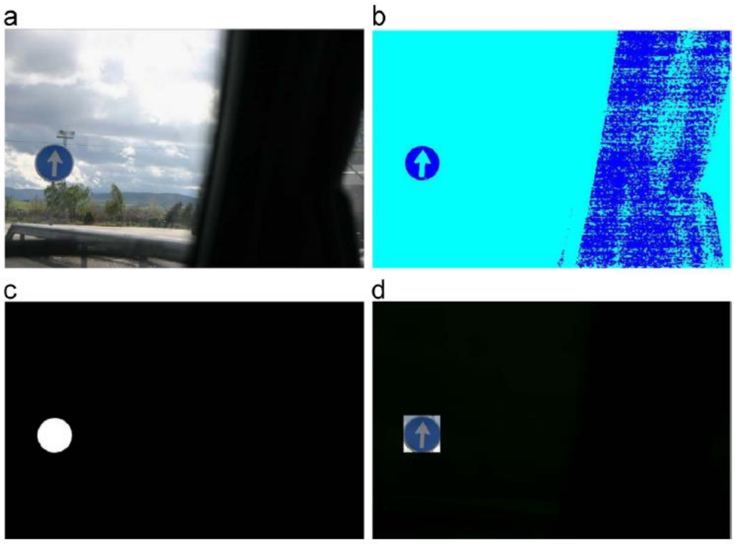
Figure 2.3.1.1

Before applying SVM classifier, this paper also proposes an area-based filtering method as a preprocessing step after image segmentation to remove erroneous regions. Given segmented blobs, the area-based filtering rules are as follows:

1) the ratio of blobs should be between 1.9 and 1/1.9;

2) the size of these blobs is restricted to 1/15 and Between 1/2 (page 4).

Segmentation result are shown below in figure 2.3.1.2



Segmentation results. (a) Original image. (b) Segmented image. (c) Candidate blobs after area-based filtering. (d) Candidate ROIs.

Figure 2.3.1.2

SVM is a data classification method based on statistical learning theory. In this paper, the authors use SVM to classify the extracted PHOG features. Four binary SVM shape classifiers have been constructed to classify the input PHOG features as triangular (label '1') or non-triangular (label '-1'), circular (label '1') or non-circular (label '-1'), inverted triangle (label '1') or non-inverted triangle (label '-1'), rhombus (label '1') or non-diamond (label '-1'). The classification outcome of a specific PHOG is determined via a voting procedure.. Thus,if the voting score is not equal to -2, the underlying ROI will be a noise region; otherwise it should be one of the traffic sign shapes.

In conclusion, this paper proposes a new traffic sign detection method and it presents a well-rounded approach to traffic sign detection, incorporating multiple techniques from image processing, shape matching, and machine learning. By utilizing color invariance, PHOG, and SVM, the method addresses the challenges of traffic sign detection under different conditions.

2.3.2 Traffic sign segmentation and classification using statistical learning methods

In this article [10], authors have proposed an automatic traffic-sign detection method capable of detecting both chromatic and achromatic signs. The proposed system devided into three stages which are:

(1) segmentation of chromatic and achromatic scene elements using L*a*b and HSI spaces, in which two machine learning techniques (k-Nearest Neighbours and Support Vector Machines) are benchmarked;

(2) post-processing to remove non-interest regions, connect fragmented signs, and distinguish signs that are located at the same post;

(3) classification of sign-shape by using Fourier Descriptors, which offers significant advantages over other contour-based methods,At the segmentation of Chromatic and Achromatic Scene Element stage is aims to segment the traffic sign regions from the background based on their color features. The method uses two color spaces: Lab and HSI. The Lab color space is used to segment the chromatic signs, which have distinct colors such as red, blue, yellow, etc. The HSI color space is used to segment the achromatic signs, which have no distinct colors such as white, black, gray, etc. The method applies two machine learning techniques to classify each pixel in the image as either traffic sign or background: k-Nearest Neighbors (kNN) and Support Vector Machines (SVM). The kNN technique assigns a pixel to the class of its k nearest neighbors in the feature space. The SVM technique finds a hyperplane that separates the classes with maximum margin in the feature space. The method compares the performance of these two techniques and finds that SVM is more accurate and robust than kNN. Figure 2.3.2.1 shows the example of Chromatic segmentation which (a)is the original image; (b) is segmented regions in for red class and (c) for blue class.

Figure 2.3.2.1

When comes to post processing stage, the method performs three post-processing operations: discarding non-interest regions, connecting fragmented signs, and separating signs located at the same post. The discarding operation removes the regions that are too small, too large, or have irregular shapes. The connecting operation merges the regions that are close to each other and have similar colors and shapes. The separating operation splits the regions that contain more than one sign at the same post. The method uses morphological operations, distance transform, watershed algorithm, and contour analysis to perform these operations. Figure 2.3.2.2 shows the

Separation of two co-located signs. (a)original RGB image; (b)segmented ROI; (c)correction of orientation; (d)filling ROI holes; (e)normalized horizontal projection and corresponding dynamics of maxima; (f)bounding box of separate sign.

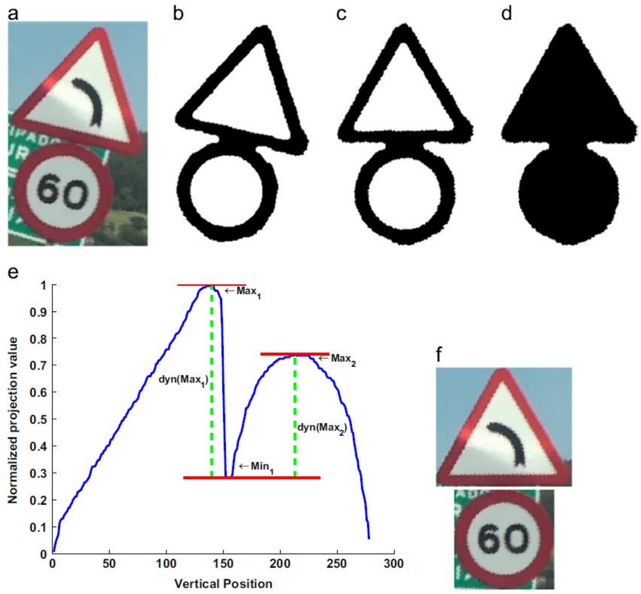


Figure 2.3.2.2

The final stage is targeted to identify the shape and category of each traffic sign region using Sign-shape classification and recognition. The method uses Fourier Descriptors (FDs) as shape features, which are invariant to scale, rotation, and translation. FDs are obtained by applying Fourier transform to the contour of each region, which converts it into a complex vector that represents the frequency components of the contour. The method uses SVM as shape classifier, which is trained with different shape classes such as circle, triangle, square, etc. The method also uses SVM as category classifier, which is trained with different category classes such as prohibitory, warning, mandatory, informative, etc. The method compares the performance of FDs with other contour-based methods such as Hu moments and Zernike moments and finds that FDs have significant advantage in terms of accuracy and robustness.

In short, the mothod proposed by the authors can achieve high accuracy, speed, and robustness in traffic sign segmentation and classification. The article also discusses some future work such as improving the color segmentation technique, incorporating more shape features, and extending the method to other types of traffic signs.

2.4 Deep Learning Method

Done by: Dai, Cheng Xiao

Traditional traffic sign segmentation methods mainly include threshold-based segmentation, image feature-based segmentation such as color-based segmentation and shape-based segmentation. In recent years, due to the rapid development of deep learning, it has become more and more important in CV tasks such as image recognition, segmentation and object detection. In this section, two methods for traffic sign segmentation based on deep learning models will be presented.

2.4.1 Traffic Signs Detection and Segmentation Based on the Improved Mask R-CNN

This paper [5] proposes an improved Mask R-CNN model for traffic sign detection and segmentation, aiming to solve the problem that it is difficult to detect and segment traffic signs due to complex backgrounds and small and medium-sized traffic signs. Additionally, a new data set TT-100K-HHU is introduced, which is an extension of TT-100K data set to solve the problem of existing data and insufficiency. This extended dataset includes more images collected from Tencent Street View and annotated using Labelme software. The improved Mask R-CNN model is trained on this extended dataset. The purpose is to provide more diverse training environments for the improved Mask R-CNN model.

The improved Mask R-CNN model contains several key enhancements. The first significant improvement being the introduction of Feature Pyramid Network [FPN] into the backbone network of Mask R-CNN. The Mask R-CNN framework consists of the backbone network, region proposal network [RPN] and the prediction network. Figure 2.4.1.1 show the Framework of Mask R-CNN. The backbone network of the improved Mask R-CNN contains parts, a bottom-up part and a top-down part. Figure 2.4.1.2 show the framework of improved Mask R-CNN. In the bottom-up part, five feature maps Ci (i = 1,2,3,4,5) are obtained. In the top-down part, transformed feature map Pi is derived from Ci and feature fusion is conducted to extend the respective fields. Compared to the backbone network of the original Mask R-CNN, the backbone network of the improved Mask R-CNN can merge high-level and low-level feature maps to include more information, small and medium-sized objects.

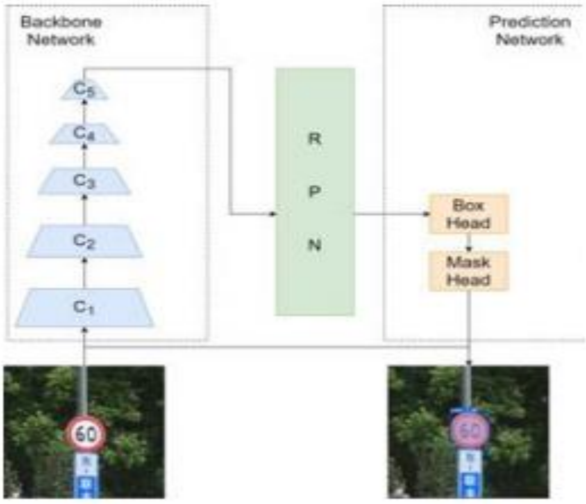


Figure 2.4.1.1 Framework of Mask R-CNN. [5]

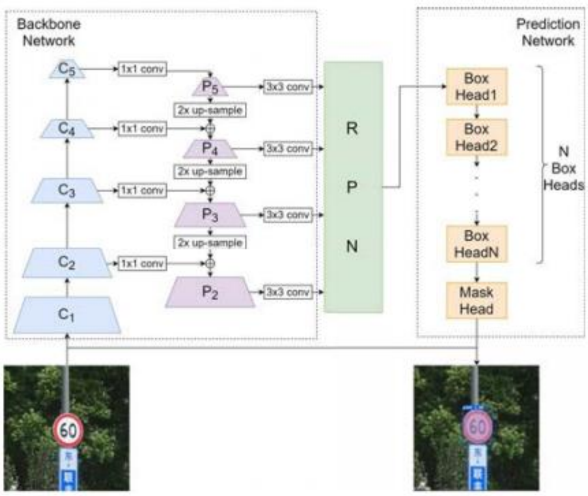


Figure 2.4.1.2 Framework of improved Mask R-CNN. [5]

As shown in figure 2.4.1.2 above, the second major improvement is the application of multiple cascaded Box Heads in the prediction network after RPN, which is used to conducting regressions to find the optimized positions. As shown in figure 2.4.1.3, the loss function L of Mask R-CNN consists of two parts. First one is the loss of the RPN, denoted as LRPN, and the second one is the loss of the prediction network, denoted as LHead. LRPN consists of classification loss Lcls and regression loss Lreg. The loss of LHead contains the multiclassification loss and regression loss of Box Head, and the pixel-wise prediction loss Lmask of Mask Head.

$$L = L_{RPN} + L_{Head}$$

$$L_{cls} = -[p_i^* log(p_i) + (1 - p_i^*) log(1 - p_i)]$$

$$L_{RPN} = \frac{1}{N_{cls}} \sum_i L_{cls} + \frac{1}{N_{reg}} \sum_i p_i L_{reg}$$

$$L_{reg} = \begin{cases} 0.5(t_i - t_i^*)^2, & if|t_i - t_i^*| < 1 \\ |t_i - t_i^*| - 0.5, & otherwise \end{cases}$$

$$L_{Head} = \frac{1}{N_{cls}} \sum_i L_{cls}^H + \frac{1}{N_{reg}} \sum_i p_i L_{reg}^H + \frac{1}{N_{mask}} \sum_i q_i L_{mask}$$

$$L_{cls}^H = -\sum_i p_i^* log(p_i)$$

$$L_{mask} = -[q_i^* log(q_i) + (1 - q_i^*) log(1 - q_i)]$$

Figure 2.4.1.3 Loss Function L of Mask R-CNN [5]

After the multiple box heads are cascaded in the prediction network for regression and the optimal position is found, the loss of the prediction network is corrected accordingly, as shown in the following figure 2.4.1.4.

$$L'_{Head} = \frac{1}{N_{cls}} \sum_i L'_{cls} + \frac{1}{N_{reg}} \sum_i p_i L'_{reg} + \frac{1}{N_{mask}} \sum_i q_i L_{mask}$$

$$L'_{cls} = \frac{1}{N}(L_{cls1}^H + L_{cls2}^H + \dots + L_{clsN}^H)$$

$$L'_{reg} = L_{reg1}^H + \frac{1}{2} L_{reg2}^H + \dots + \frac{1}{2^{N-1}} L_{regN}^H$$

Figure 2.4.1.4 Loss Function of Prediction Network for Cascaded Box Heads. [5]

At the same time, after testing different backbone networks and different numbers of Box Heads and the IoU thresholds, the authors concluded that adopting ResNet-50 with FPN as the backbone network, cascading 3 Box Heads in the prediction network and using data augmentation methods like image flipping, image cropping, noise addition and similar techniques will achieve the best accuracy. Figure 2.4.1.5 show the result of Ablation experiments for the developed network.

Network		ResNet-50-FPN	ResNet-50-FPN with data augmentation	Improved Mask R-CNN	Improved Mask R-CNN with data augmentation
Task:bbox	mAP(0.5:0.95)	0.654	0.674	0.714	**0.738**
	mAP50	0.827	0.844	0.896	**0.915**
	mAP75	0.779	0.803	0.854	**0.876**
	mAPs	0.318	0.323	0.365	**0.385**
	mAPm	0.653	0.659	0.715	**0.745**
	mAPl	0.791	0.825	0.844	**0.857**
Task:mask	mAP(0.5:0.95)	0.592	0.607	0.631	**0.644**
	mAP50	0.784	0.797	0.846	**0.866**
	mAP75	0.691	0.700	0.740	**0.756**
	mAPs	0.255	0.260	0.305	**0.314**
	mAPm	0.599	0.610	0.648	**0.671**
	mAPl	0.716	0.747	0.740	**0.753**

Figure 2.4.1.5 Ablation experiments for the developed network [5]

To sum up, As shown in Figure 2.4.1.6, the improved Mask R-CNN model proposed in [5] demonstrates better performance in detecting and segmenting small and medium-sized traffic signs from actual street view collected images. However, for traffic signs with complex poses or occlusions, the segmentation accuracy of the model can be further optimized, and more different data augmentation methods can be explored to increase the robustness of the model to cope with complex scenes. At the same time, as the authors proposed in the paper, the infer speed can be reduced by reducing the model size, so that the model can be deployed to the mobile terminal with limited computing power, and the applicability of the model in the actual scene can be increased.

(a) Mask R-CNN with ResNet-50 (b) Mask R-CNN with ResNet-50-FPN (c) The improved Mask R-CNN (a) Mask R-CNN with ResNet-50 (b) Mask R-CNN with ResNet-50-FPN (c) The improved Mask R-CNN

Figure 2.4.1.6 Results comparison before and after the improvements *[5]*

2.4.2 Automatic Traffic Sign Detection and Recognition Using SegU-Net and a Modified Tversky Loss Function With L1-Constraint

In this paper [6], traffic sign segmentation and detection are treated as an image segmentation problem and a deep convolutional neural network based method is proposed to solve it. The authors propose a new network named SegU-Net to detect traffic signs from video sequences by merging the cutting-edge segmentation architectures SegNet and U-Net. The network was tested on the German Traffic Sign Detection Benchmark [GTSDB]dataset, and the precision and recall reached 95.29% and 89.01%, respectively. Figure 2.4.2.1 show the Performance on GTSDB. Comparing the performance of the proposed SegU-Net segmentation network with two state-of-the-art segmentation networks PSPNet and DeepLabv3, evaluating only the segmentation performance of the network, we will find that Seg-Net has an average IoU of 0.53, However, the average IOU of PSPNet and DeepLabv3 are 0.26 and 0.35, respectively.

	Proposed SegU-Net with VGG-like Classifier		Faster R-CNN Inception Resnet V2 [61]		R-FCN Resnet 101 [61]	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Prohibitory	95.48	91.93	96.99	100.00	84.66	99.38
Mandatory	92.31	73.47	79.31	93.88	76.67	93.88
Danger	96.72	93.65	92.19	93.65	86.76	93.65
Overall	95.29	89.01	92.36	97.44	83.60	97.07

Figure 2.4.2.1 Performance on GTSDB [6]

Firstly, as shown in Figure 2.4.2.2 below, the authors adopt a modular approach to design their process, which allows them to handle different tasks of traffic sign detection independently. In this paper [6], the authors propose a model based on CNN to detect and recognize traffic signs. The model consists of two modules. The first module is a CNN, called Localizer, which is responsible for segmenting and detecting traffic signs from the image. The second module is a classifier, which is responsible for identifying the traffic signs detected by the Localizer.

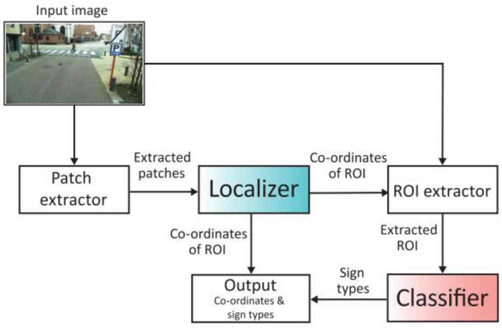


Figure 2.4.2.2 Outline of the model proposed for traffic sign detection [6]

This approach allows the authors to use different optimization schemes in different parts of the process. For example, they can add challenge-aware pre-/ post-processing, denoising, etc. modules without the need to fine-tune the whole model. Because each network can be optimized and trained independently without affecting the other networks. And in a constrained hardware environment, it is much easier to train several deep networks to handle each specific task than to train a single but very deep network to handle the entire system.

In fact, the Localizer is implemented by SegU-Net. This is a Fully Convolution Neural Network architecture that connects the encoder as the down sampling stage and the decoder as the up sampling stage through a residual learning strategy. The localizer is responsible for isolating possible traffic sign areas from the image. The location of the object is produced by performing pixel-wise image segmentation. Figure 2.4.2.3 show The detailed SegU-Net architecture.

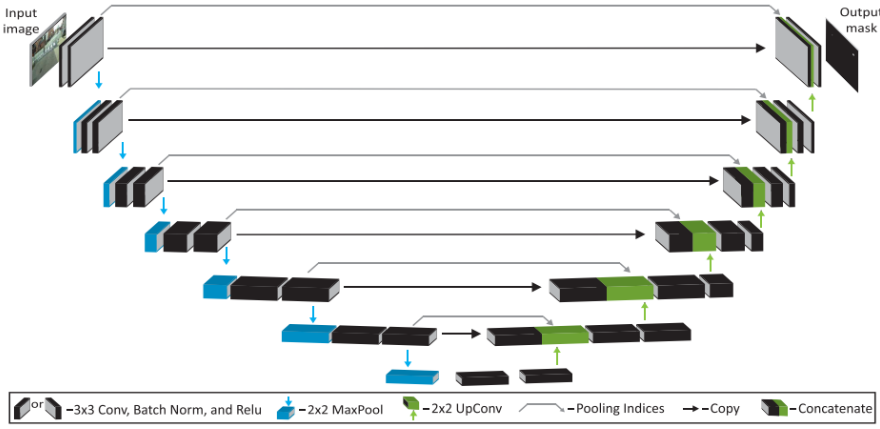


Figure 2.4.2.3 The detailed SegU-Net architecture [6]

The SegU-Net merges SegNet and U-Net, enhancing small sign detection by combining their benefits. U-Net is robust against low resolution features and uses multiple feature channels for localizing objects in noisy conditions. SegNet uses up sampling in the decoder stage, enabling more accurate segmentation of pixels than conventional up convolution. This combination enables the network to detect small traffic signs with greater confidence and pinpoint their location more accurately than standalone architectures. The combination of these two techniques enables the model to effectively segment traffic signs from images for subsequent recognition and classification.

Secondly, in order to train SegU-Net, the authors use Tversky loss function and add L1 constraint instead of intersection/union loss which is traditionally used to train segmentation networks. This loss function allows the weights of false positives and false negatives to be adjusted during training. Figure 2.4.2.4 show the Tversky loss function. Also, figure 2.4.2.5 shows the effect of using L1 constraint.

$$L(P,G,\alpha,\gamma) = 1 - \frac{|P \cap G|}{|P \cap G| + \alpha \cdot |P - G| + \beta \cdot |G - P| + \epsilon} + \gamma \cdot \sum_{i,j} |P_{ij} - G_{ij}|$$

Figure 2.4.2.4 Show the Tversky loss function [6]

Model	True Positive	False Positive	False Negative	Precision (%)	Recall (%)
U-Net (IoU Loss)	5874	2139	5670	73.30	50.88
SegU-Net (IoU Loss)	8335	420	3209	95.20	72.20
SegU-Net (Tversky Loss)	9136	654	2245	93.31	80.27
SegU-Net (Tversky Loss + L1 Constraint)	**9104**	**520**	**2246**	**94.60**	**80.21**
SegU-Net (Tversky Loss + L1 Constraint) Resized Input	8452	649	2872	92.87	74.64

Figure 2.4.2.5 Performance metrics for each progressive association of the test set with the localizer network for the CURE-TSD dataset [6]

To summarise, the traffic sign segmentation technique used in the paper [6] is a deep learning based approach that primarily uses the SegU-Net and an improved Tversky loss function, in conjunction with a modular flow design. In the experiments, SegU-Net outperforms other networks, which further justifies the choice of SegU-Net as the segmentation network. It is also tested on different datasets such as GTSDB and CURE-TSD to prove its generality. However, it is still possible to improve the training strategy, loss function and network architecture to robustly perform detection and recognition tasks on more complex datasets.

2.5 Comparison of the 4 Techniques

In the literature review above, we have summarised four types of methods to achieve traffic sign segmentation, each of which has its own unique advantages and application scenarios.

The traditional methods in Section 2.1 are simple and intuitive for traffic sign segmentation in simple scenarios, but have limited effect in traffic sign recognition in complex scenarios and diversity. The parameters need to be adjusted manually, thresholds and rules need to be reset for new traffic sign categories, and the effect may be limited for signs with inconspicuous colours or large-scale data.

The machine learning methods in Section 2.2 are able to perform classification and segmentation by learning sample data, and Support Vector Machine [SVM] have strong robustness and generalisation capabilities and perform well in some simple scenarios. However, SVM may require a large amount of training data and computational resources, and may face the problem of computational overhead and insufficient training data for traffic signs with inconspicuous colours or large-scale data.

The hybrid traditional and machine learning approach in Section 2.3 effectively detects traffic signs in complex scenarios by combining the advantages of both traditional and machine learning approaches. The SVM classifier is able to learn sample data, which improves the accuracy of classification and segmentation. However, the segmentation accuracy may be limited for traffic signs with complex poses or occlusions.

The deep learning methods in Section 2.4 are able to learn feature representations from large amounts of data with strong adaptive and generalisation capabilities. The improved Mask R-CNN and SegU-Net perform well in traffic sign segmentation and recognition, with better performance for small-sized traffic signs. However, it is more demanding for model training and tuning, requiring larger datasets and computational resources. Meanwhile, for traffic signs in complex scenes, the segmentation accuracy may need to be optimised.

In summary, various methods have their advantages and limitations. Traditional methods are simple and easy to use, but perform poorly in complex scenarios; machine learning methods have strong generalisation ability, but require a large amount of training data and computational resources; hybrid methods combine the advantages of traditional and machine learning, and are more effective in some scenarios; deep learning methods are able to learn feature representations from a large amount of data, but have high requirements for training and tuning of the model. There is no absolute best or worst of these methods, we should choose the most suitable method in different scenarios and conditions.

Chapter 3

3.1 Design Specifications

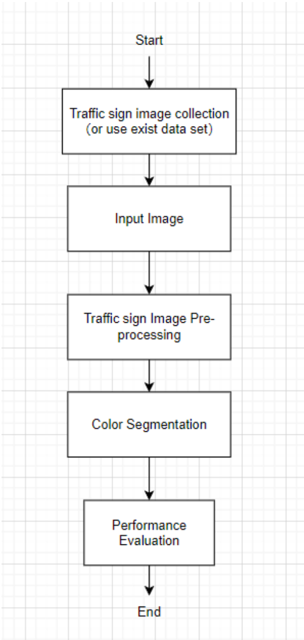


Figure 3.1.1 Block diagram

Image preprocessing such as scaling and brightness correcting are necessary steps before performing the traffic sign segmentation in order to eliminate out the the quality of the images as the images captured (dataset) might low quality due to the environment factor such as the surrounding brightness, contrast level, weather condition and the images capturing from different angle.

In addition, the segmentation approach we proposed to use in this paper is colour segmentation which the RGB traffic sign images will be first perform the color conversion to HSV so to make the images having 3 channels (Hue, Saturaion, Brightnmess) to perform the preprocessing stage. Lastly, performance will be evaluated to ensure the accuracy.

3.2 System Design/Overview

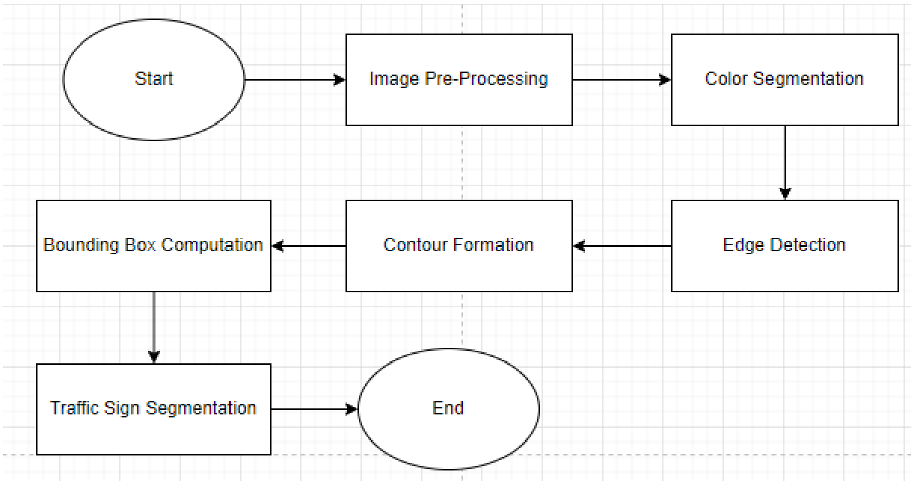


Figure 3.1.2 System Design diagram

3.2.1 Image Pre-processing

The input dataset will undergoes several prepocessing method to improve the suitability and prepare for the color segmentation. There are few substep perfomed which are:

a. Image Resizing: The input image is resized to a standardized dimension and aid in achieving a better resolution of the image.

c. Color Space Conversion: Image is converted from HSV to different color space such as HSV(Hue, Saturation, Value) which represent type of colors based on the degree or the color , intensity of color in the images and brightness of color in the image .This process can aid the color segmentation stage as it can direct detect the pixels of the image fall on the specified range.

3.2.2 Color Segmentation

A technique to isolate the regions of traffic sign based in color characteristics.

a. Color Thresholding: define specific ranged of color in chosen color space. This process generates a binary image where pixels detected correspond to specified range will be marked as white, otherwise will be marked as black.

3.2.3 Edge Detection and Contour Formation

This step performs a series of operations on the colour mask to better identify the edges and contours of the object.

At first, a binarization operation is performed to convert the colour mask into a binary image where the object region is white and the background region is black. Then, the binary image is processed using morphological operations (erosion and dilation) to decrease noise and improve the quality of contours.

Subsequently, the Canny edge detection algorithm is used to find edges on the processed image. Finally, the contours in the image are found using the getContours() function, which usually selects the contour with the largest area.

3.2.4 Largest Contour Selection

In this step, the contour with the largest area is selected from all detected contours, which usually corresponds to the contour of the main object.

3.2.5 Bounding Box Computation

Once the largest contour is found, we use OpenCV's cv.boundingRect() function to compute the smallest rectangular bounding box enclosing the contour based on its coordinates.

3.2.6 Intersection over Union (IoU) Calculation

In this step, IoU computation is done by comparing the area of intersection between the predicted and real bounding boxes with their concatenated area. This IoU value provides an important metric for measuring the accuracy of object detection by telling us the degree of overlap between the detected bounding box and the actual object position.

3.2.7 Results Categorization

In this step, we classify the detection results for each image based on the calculated IoU (intersection and concurrency ratio) value. If the IoU is greater than or equal to 0.55, the result is classified as a correct detection result and recorded in the trueTotal list. If the IoU is less than 0.55 and the predicted bounding box width is 0 (indicating that no object was detected), classify the result as uncertain and add it to the zero list. For other cases, the result is classified as an incorrect detection and recorded in the falseTotal list. In addition, the detection accuracy and the time taken to perform the entire detection process are also calculated and output.

3.3 Implementation Issues and Challenges

During the implementation phase of the proposed project, it is inevitable to encounter challenges. The first problem we need to face is the limitations and deficiencies of the Traffic Sign Recognition Database [TRSD], the image quality of TRSD is low, the image size is not uniform and there are problems such as noise, blurring and occlusion, which means that we need to preprocess and normalise the images. In addition, the traffic sign categories in the dataset are unevenly distributed and the labelling information is incomplete. Some categories have many samples, but some categories have only a few samples. In addition, TRSD does not provide attributes such as rotation angle, shape and colour of the signs. These shortcomings can limit the ability of our method to segment the details of traffic signs. This means that in future work we may need to combine a variety of features other than colour, and for the lack of colour information, we can try to convert the image to other colour spaces, such as HSV, LAB, etc., to better capture the colour features of the sign and thus improve the accuracy of segmentation.

Secondly, another challenge is the complexity of the image background and the influence of weather factors, light factors. These factors can cause the visibility of traffic signs in the image to decrease and increase the difficulty level of the segmentation algorithm. This makes it necessary to perform operations such as image enhancement, colour correction, etc. to improve the image quality and visibility of the traffic signs.

Chapter 4

4.1 Hardware Setup

Description	Specifications
Model	HP EliteBook 845
Processor	AMD Ryzen 9 PRO 6950HS
Operating System	Windows 11
Graphic	AMD Radeon(TM) Graphics
Memory	16GB DDR4 RAM
Storage	1TB SSD

Table 4.1 Specifications of laptop

4.2 Software Setup

1. Jupyter Notebook

Version: 6.5.4

It is a powerful web-based application that allows the user to execute the their code on the browser and visualize the data as it can integrate with the some famous library such as Matplotlib, Pandas, NumPy and it does support programming language such as Python, R language. In this project, we have chosen the Python to run on this platform.

2. OpenCV

Version: 4.7.0

It is an open-source computer vision library that provides a set of tools and functions for processing images and videos. The library is used extensively in the field of computer vision and image processing for tasks such as image processing, object recognition, feature extraction, image segmentation, face recognition, camera calibration, etc. With features that include cross-platform support for a variety of programming languages such as C++, Python, and Java, OpenCV is an important tool in the field of image processing.

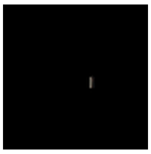
Chapter 5: System Evaluation and Discussion

5.1 System Testing and Performance Metrics (dai)

Original Image	Segmented Image	IoU Value
		**0.60**
		**0.68**
		**0.57**
		**0.59**
		**0.64**
		**0.94**

		**0.94**
		**0.44**
		**0.84**
		**0.94**
		**0.60**
		**0.00**

		**0.64**
		**0.94**
		**0.44**
		**0.91**
		**0.82**
		**0.61**

		**0.62**
		**0.65**
		**0.68**
		**0.62**
		**0.01**
		**0.64**

		**0.63**
		**0.20**
		**0.06**
		**0.73**
		**0.94**
		**0.96**

		**0.63**
		**0.73**
		**0.89**
		**0.51**
		**0.68**
		**0.92**

		**0.86**
		**0.90**
		**0.94**
		**0.21**
		**0.92**
		**0.67**

		**0.73**
		**0.92**
		**0.87**
		**0.90**
		**0.62**
		**0.87**

		**0.67**
		**0.57**
		**0.93**
		**0.90**
		**0.58**
		**0.95**

		**0.96**
		**0.57**
		**0.97**
		**0.63**
		**0.98**
		**0.49**

		**0.60**
		**0.97**
		**0.98**
		**0.94**
		**0.98**
		**0.17**

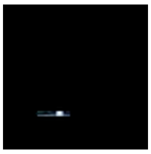
		**0.00**
		**0.64**
		**0.91**
		**0.92**
		**0.62**
		**0.61**

		**0.68**
		**0.94**
		**0.97**
		**0.98**
		**0.95**
		**0.92**

		**0.97**
		**0.93**
		**0.75**
		**0.22**
		**0.97**
		**0.96**

		**0.99**
		**0.63**
		**0.91**
		**0.94**
		**0.97**
		**0.99**

		**0.92**
		**0.97**
		**0.97**
		**0.92**
		**0.99**
		**0.95**

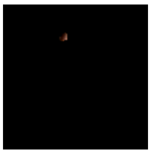
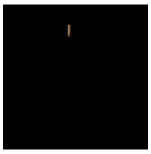
		**0.95**
		**0.00**
		**0.98**
		**0.35**
		**0.94**
		**0.38**

		**0.92**
		**0.92**
		**0.89**
		**0.85**
		**0.84**
		**0.44**

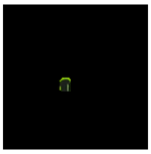
		**0.55**
		**0.00**
		**0.96**
		**0.97**
		**0.00**
		**0.97**

		**0.96**
		**0.95**
		**0.00**
		**0.98**
		**0.00**
		**0.00**

		**0.00**
		**0.00**
		**0.00**
		**0.00**
		**0.00**
		**0.86**

		**0.00**
		**0.00**
		**0.07**
		**0.01**
		**0.75**
		**0.00**

		**0.93**
		**0.93**
		**0.94**
		**0.00**
		**0.02**
		**0.66**

		**0.17**
		**0.61**
		**0.02**
		**0.00**
		**0.00**
		**0.01**

		**0.00**
		**0.84**
		**0.00**
		**0.93**
		**0.85**
		**0.84**

		**0.00**
		**0.12**
		**0.13**
		**0.12**
		**0.97**
		**0.96**

		**0.65**
		**0.97**
		**0.95**
		**0.62**
		**0.57**
		**0.60**

		**0.17**
		**0.98**
		**0.97**
		**0.98**
		**0.50**
		**0.51**

		**0.97**
		**0.99**
		**1.00**
		**0.94**

5.2 Testing Setup and Result

IOU (Intersection of Union)	Number of Images fulfil the IOU
> 0.55	131
> 0.70	95
> 0.90	74

In order to evaluate the proposed model of this project in traffic sign segmentation, IOU is applied as it's popular to use to evaluate the performance of object segmentation in terms of accuracy which calculated by measuring the overlapping section of the actual bounding box (ground truth) with the predicted bounding box.

According to the table, our model has successfully segmented out 131 traffic sign in the dataset out of 171 images with an IOU value of 0.55. When come to an IOU value of 0.70, 95 traffic sign has been segmented out. Lastly, when the IOU threshold set to 0.90, only 74 traffic sign images out of 172 iamges were segmented out successfully.

5.3 Error Analysis (Paun Tek You)

There are a few challenges have been identified when implementing this project. One of the challenges is that the proposed model is difficult to segment out the traffic sign when it has a complex background. For example, traffic sign images with vehicle as background capture by non-professional device and appear another traffic sign in the same image.

Furthermore, there are still of the traffic sign unable to segmented out from the images due to the quality of the image present in the dataset, some of the images are low in image resolution as shown in figure 5.3.x.x although resizing is done in the preprocessing stage but it's not the optimal method for all images. In addition, some of the traffic sign is cover by the obstacles causing the proposed segmentation process unsuccessful.

Moreover, the proposed segmentation model is applying the static thresholding by defining the HSV value, this has caused the segmentation result not optimal for all kind of traffic images due to the colour variation and environmental issue which means there are a chance the pixel value is not fall in the defined range.

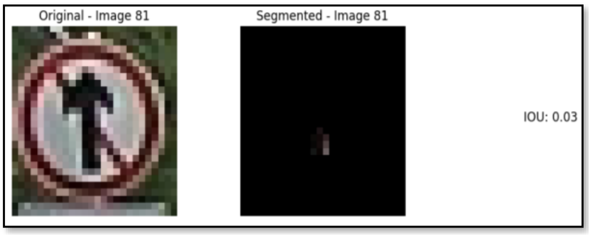


Figure 5.3.x.x: Blur Images

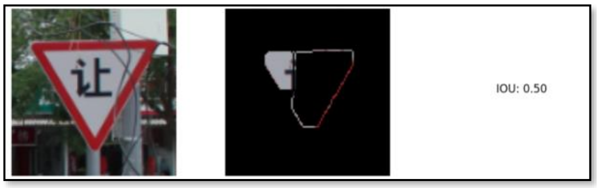


Figure 5.3.x.x: Images with obstacle

5.6 Compare with previous work.

To establish a basis for comparison, we consider the findings from the paper titled "Image Segmentation and Shape Analysis for Road-Sign Detection," as described in Chapter 2. While the paper did not explicitly report the segmentation step's accuracy, it presented results indicating the successful detection of 92.3% of road signs in a test set comprising 100 images obtained from Yahoo Image, with varying environmental conditions and traffic sign types.

However, it is important to note that this reported accuracy encompasses not only the segmentation but also the subsequent steps of shape analysis and recognition. Thus, a more accurate evaluation of the segmentation step requires a comparison of the segmented regions of interest (ROIs) with manually annotated ground truth ROIs created by human experts.

In contrast to the approach outlined in the previous study, we adopted a traditional methodology for traffic sign segmentation. Our experiment utilized a dataset of traffic sign images from the Chinese Traffic Sign Database, which contains 457 images with various environmental conditions and traffic sign types. Our experiment resulted in an overall accuracy of 72.09% based on the IOU metric. An IOU value of 0.55 or higher was considered indicative of successful segmentation.

Upon comparing our findings with those of the previous study, we note a significant disparity in reported accuracy:

- **Our Experiment (Segmentation Accuracy)**: 72.09%
- **Previous Study (Indirect Estimate)**: Approximately 92.3%

It is crucial to exercise caution when interpreting these results due to several factors that may influence the comparison's fairness:

1. Differences in Datasets: Our experiment utilized a dataset of traffic sign images with distinct characteristics and environmental conditions compared to the dataset in the previous study. Datasets play a pivotal role in performance evaluation.
2. Evaluation Metrics: The previous study's accuracy was derived from overall road sign detection, which includes shape analysis and recognition stages. In contrast, our accuracy specifically assesses the segmentation stage using IOU.
3. Experimental Settings: Variations in experimental setups, such as preprocessing parameters or algorithmic configurations, could affect the outcomes.

Given these considerations, we acknowledge that our segmentation accuracy falls short of the indirectly estimated accuracy from the previous study. To enhance the effectiveness of our method, we propose several avenues for improvement in the next section, aiming to bridge the gap and achieve more competitive results in future experiments.

Chapter 6: Conclusion and Recommendation

6.1 Conclusion

(done by Lee Jian Shen)

In short, the project achieved the goal of identifying and isolating traffic sign images by simulating various traffic sign scenarios, including different angles, image quality, and blurring. The system achieved an accuracy rate of 76%, despite encountering difficulties such as low image resolution and some traffic signs being obstructed, which caused occasional segmentation failures. Nevertheless, this project success opens the door for further improvements. For instance, after successfully extracting the traffic sign from the input image, we may now think about implementing object categorization and recognition.

6.2 Recommendations

(done by Lee Jian Shen)

The current system is able to be identifying traffic sign images in basic background, but it might be a improvement for more complex details. To increase the system robustness across multiple images sets, future work should involve the implementation of advanced algorithms and exploring different preprocessing methods.

.

References

[1] M. Swathi and K. V. Suresh, "Automatic traffic sign detection and recognition: A review," *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, Chennai, India, 2017, pp. 1-6, doi: 10.1109/ICAMMAET.2017.8186650.

[2] N. Mohd Ali, M. S. Karis, A. F. Zainal Abidin, B. Bakri, E. F. Shair, and N. R. Abdul Razif, "TRAFFIC SIGN DETECTION AND RECOGNITION: REVIEW AND ANALYSIS", Jurnal Teknologi, vol. 77, no. 20, Dec. 2015.

[3] S. B. Wali et al., "Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges," Sensors, vol. 19, no. 9, p. 2093, May 2019, doi: 10.3390/s19092093.

[4] C. Timbuş, V. Miclea and C. Lemnaru, "Semantic segmentation-based traffic sign detection and recognition using deep learning techniques," 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 2018, pp. 325-331, doi: 10.1109/ICCP.2018.8516600.

[5] H. Qian, Y. Ma, W. Chen, T. Li, Y. Zhuo and W. Xiang, "Traffic Signs Detection and Segmentation Based on the Improved Mask R-CNN," 2021 40th Chinese Control Conference (CCC), Shanghai, China, 2021, pp. 8241-8246, doi: 10.23919/CCC52363.2021.9549552.

[6] U. Kamal, T. I. Tonmoy, S. Das and M. K. Hasan, "Automatic Traffic Sign Detection and Recognition Using SegU-Net and a Modified Tversky Loss Function With L1-Constraint," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1467-1479, April 2020, doi: 10.1109/TITS.2019.2911727.

[7] "Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry," Future Generation Computer Systems, vol. 94, pp. 381–391, May 2019, doi: https://doi.org/10.1016/j.future.2018.11.027.

[8] J. F. Khan, S. M. A. Bhuiyan and R. R. Adhami, "Image Segmentation and Shape Analysis for Road-Sign Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 1, pp. 83-96, March 2011, doi: 10.1109/TITS.2010.2073466.

[9] H. Li, F. Sun, L. Liu, and L. Wang, "A novel traffic sign detection method via color segmentation and robust shape matching," Neurocomputing, vol. 169, pp. 77–88, 2015. doi:10.1016/j.neucom.2014.12.111

[10] J. M. Lillo-Castellano, I. Mora-Jiménez, C. Figuera-Pozuelo, and J. L. Rojo-Álvarez, "Traffic sign segmentation and classification using Statistical Learning Methods," Neurocomputing, vol. 153, pp. 286–299, 2015. doi:10.1016/j.neucom.2014.11.026

[11] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez and S. Lafuente-Arroyo, "Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition," in IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 4, pp. 917-930, Dec. 2010, doi: 10.1109/TITS.2010.2054084.

[12] V. R.Deshmukh, G. K. Patnaik, and M. E. Patil, "Real-Time Traffic Sign Recognition System based on Colour Image Segmentation," *International Journal of Computer Applications*, vol. 83, no. 3, pp. 30–35, Dec. 2013, doi: https://doi.org/10.5120/14430-2575.