Final Report: Detection of traffic signs

Loïc Le Bescond, Abderrahim Mehdaoui, Adnan Zeddoun
CentraleSupélec, Option Mathématiques Appliquées
{loic.lebescond, abderrahim.mehdaoui, adnan.zeddoun}@{supelec,student.ecp}.fr

1. Abstract

Detection and recognition of road traffic signs are an important element and holds strong issues such as in Advanced Driver Assistance Systems (ADAS) and autonomous vehicles, which can provide real-time road sign perception information to vehicles. We propose here to implement a traffic sign detection method proposed by [10] which is based on adaptive color threshold segmentation, MSER algorithm and the hypothesis of shape symmetry.

First, we designed an approximate maximum and minimum normalization method, which is used to suppress background and interference of high brightness area in image. Based on this, we have then calculated an adaptive segmentation threshold using the cumulative distribution function of the image histogram to fix the threshold. After that, we turn the highlight shape feature of thresholding image into a connected domain feature vector. Finally, we tried to efficiently extract the regions of interest (ROI) of traffic signs by implementing a shape symmetry detection algorithm based on statistical hypothesis testing. We then performed some experiences on the GTSDB (German Traffic Sign Detection Benchmark) dataset to have some empirical results.

The accuracy of traffic sign detector that we have implemented didn't exceed that of the reference work. However, we could find some good results on some images.

2. Introduction

Here we seek to study and implement a method for traffic signs detection. This problem has been widely studied in the scientific literature because of the potential economic and social impact it represents. Amongst others, we can quote the design of systems for autonomous vehicles or driver assistance systems. Traffic signs are usually composed of specific shapes (round, square, and triangular...) and colors (red, blue, and yellow...), which have strong visual effect in the road environment (they have been designed for this specific purpose). Therefore, traffic sign detection methods can be classified into three categories, color-based, shape-based and hybrid (color-shape-based) methods.

In color-based traffic sign detection, the RGB images are usually transformed into another color space. Then the traffic signs are extracted using color thresholding segmentation. Color-based traffic sign detection methods have some weaknesess, for instance, they are easily affected by complex illumination conditions of traffic scene... So under poor conditions the accuracy might no be as good as we want.

In shape-based traffic sign detection, we try to detect traffic signs by using their shape properties. For example, we can use geometric invariant moment and geometric symmetry or even template matching. In a complex illumination environment, geometric symmetry has better adaptability compared to template matching or geometric invariant moment, but it requires higher computational complexity.

The existing color and shape-based traffic sign detection methods have weaker adaptability under complex brightness conditions. We propose to implement here an adaptive color threshold segmentation and highly efficient shape symmetry algorithm to achieve robust traffic sign detection in a complex illumination environment.

The report is organized as follows: Section 2 gives a more formal and shorter definition of the problem, section 3 describes related works about traffic sign detection, Section 4 details the general methodology we used, Section 5 explains how we evaluate our works and some experimental results, then Section 5 presents the conclusions.

3. Problem definition

We will try to detect traffic sign by implementing a method which takes in account the shape and colors properties of traffic sings.

In our study we will consider these notations:

Notations for MSER Algorithm

- $I : (x, y) \in \mathrm{D} \subset \mathbf{N}^2 \mapsto I(x, y) \in S = \{0, 1, ..., 255\}$ (only one dimension after preprocesses, step 1) represents the function that maps coordinate (x, y) into intensity $I(x, y)$ in the image. By abuse of notation, we will also consider I as an image
- $\mathbf{p}, \mathbf{q} \in \mathbf{N}^2$ represent two pixels in image I
- We note A an adjacency relation in $\mathrm{D} \times \mathrm{D}$. $\mathbf{p}A\mathbf{q}$ means $\mathbf{p}$ and $\mathbf{q}$ are neighbors in the image
- Q is a contiguous subset of D when:
$\forall\, \mathbf{p}, \mathbf{q} \in Q, \exists\, n \in \mathbf{N}^*, a_1, ...a_n \in Q$ such as:
$\mathbf{p}Aa_1$, a_1Aa_2,...,$a_{n-1}Aa_n$, $a_nA\mathbf{q}$
- We define region boundary of Q as:

$$\partial Q = \{q \in D \backslash Q : \exists\, p \in Q, pAq\} \tag{1}$$

- We call a maximum intensity region Q a region of D such as:

$$\forall\, p \in Q, q \in \partial Q,\ I(p) > I(q) \tag{2}$$

Notations for Statistical Filtering and geometrical constraints

- We will note $C = [C_1, ..., C_i, ..., , C_N]$ the vector containing the number of pixels per column C_i, d the vector of features that we will construct to assess whether the region is symmetric or not. We will then suppose that the d_k are independent and identically distributed, following a $\mathcal{N}(\mu, \sigma)$ law.
- We will note w_{min} and w_{max} are the minimum and maximum allowable widths for our bounding boxes and h_{min} and h_{max} are the minimum and maximum heights.

Objective

Our goal is to find rectangular Bounding Boxes which fit as well as possible Traffic Signs in our image I. To do this, our method finds contiguous subsets $(Q)_{i=1}^m$ in the image I (m is the number of subsets detected which is ideally equal to the number of traffic signs). Each subset Q_i is composed by pixels.
When the contiguous subsets are found by our method, we find the smallest rectangular bounding boxe that surround each subbset.

Then these bounding boxes $(B_{detected_i})_{i=1}^m$ are compared to the ground truth bounding boxes $(B_{GT_i})_{i=1}^m$ with the Jaccar Similarity index for each image (by abuse of notation we consider that we have exactly the same number of detected bounding boxes and ground truth bounding boxes): $\forall i \in \{1, \ldots, m\}$,

$$J(B_{detected_i}, B_{GT_i}) = \frac{|B_{detected_i} \cap B_{GT_i}|}{|B_{detected_i} \cup B_{GT_i}|} \tag{3}$$

The objective of Traffic Sign Detection is to maximize this Jaccar Similarity index.

4. Related works

As mentioned earlier, traffic signs usually have a strict color scheme and specific shapes. So, the detections methods find in the literature can be classified into three categories, color-based, shape-based and hybrid methods. We are focused on hybrid methods, however, having an idea of what has be done in the literature for each class might be very helpful to understand our project.

For color-based methods, using size and perspective invariability [9] of traffic signs' color can be very helpful in traffic signs detection. In fact, the ROI of traffic signs can be quickly located based on color information. They generally use color threshold segmentation in a specific color space to extract the ROIs. Escalera [8] sets the thresholds of RGB color channels to perform segmentation of RGB images. Brightness conditions of a complex traffic environment can has bad effects on the segmentation since RGB color space is sensitive to brightness. Therefore, there are methods to pass the images from RGB color space to other color space. Creusen [5] extracted the HOG feature on YcbCr color space to detect traffic signs. Greenhalgh [6] proposed the Red–Blue preprocessing method to perform color normalization on the red and blue channels of images to reduce the influence of complex brightness conditions.

Shape information represents another well-known and prominent feature of traffic sign. Contour features are usually used for detection of traffic sign's ROI. F. Moutard [3] detected round and rectangular traffic signs by Hough transform and the well-known Canny edge detection. We also find obvious geometrical symmetry on traffic signs. For example, Xu [13] used shape symmetry detection methods to extract circular and triangular signs. Wang [12] detected radial symmetry based on local affine invariant edge response. Although shape information-based traffic sign detection is more adaptable to complex brightness conditions, it requires higher computing complexity and the performance is influenced greatly by deformation, rotation, and partial occlusion of traffic signs.

Color and shape features are combined into traffic sign detection algorithms to enhance detection performance in an hybrid method. Deguchi [2] used a classifier which is based on local rank pattern feature to classify RGB pixel values to obtain edge images for seven traffic signs, and then used RANSAC circular fitting to detect circular traffic signs. The detection method proposed in [4] [11] had better robustness in a complex road environment. However, the majority of these methods are only appropriate for a specific category of traffic signs. For instance, they detected only red circular signs in [4] [11]. Furthermore, these methods were usually tested on different dataset created by the authors rather than public traffic sign datasets.

So, the existing color and shape-based traffic sign detection methods don't have well behavior under complex brightness conditions. Hence, we propose to implement here a novel adaptive color threshold segmentation and statistical hypothesis test symmetry detection algorithm to enhance traffic sign detection performance under complex illumination environments.

5. Methodology

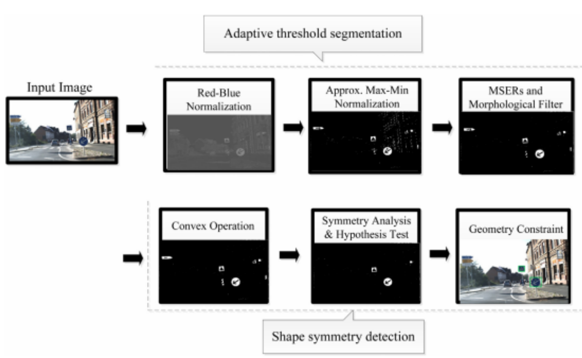

Figure 1. Detection process

5.1. Convert RGB images into Normalized Red-Blue images

First, as we consider blue and red traffic signs, we convert our RGB images (3 dimensions, one for each color) into normalized red/blue images. This transformation is defined for each pixel:

$$\Omega_{RB} = max\left(\frac{R}{R+G+B}, \frac{B}{R+G+B}\right) \tag{4}$$

So the resulting pixel values in this image are higher for red and blue pixels and lower for other colors. MSER regions are then found for these new images. An example of the normalized red/blue transform and the thresholding applied to the image after normalization are shown in Illustration part . We can see that the red and blue road signs maintain their shape at several threshold levels, making them valuable candidates for detection.

5.2. Adaptative thresholding using empirical cumulative distribution

After the red-blue normalization, we apply a thresholding segmentation to our images: we calculate an adaptive threshold m_1 by using the cumulative distribution function of image histogram. In fact, they have deduced empirically that the foreground region's gray values are often distributed within a certain range. So we choose m_1 as the quantile of order 90%. Then we use the approximate maximum–minimum normalization method to process images according to the threshold.

$$I(\mathbf{p}) = \begin{cases} 0 & \text{if } RB(\mathbf{p}) < m_1 \\ 255 & \text{if } RB(\mathbf{p}) = m_1 \\ \frac{RB(\mathbf{p})-m_1}{\max(RB)-m_1} \times 255 & \text{if } RB(\mathbf{p}) > m_1 \end{cases} \tag{5}$$

With this method, the interference of overexposure foreground objects and background is suppressed, so we can achieve better traffic sign segment effect in complex lighting environments. The threshold has been find by observing that the gray values of the foreground region are often distributed within a certain range by statistical analysis experiments

5.3. Morphological filtering

Even after these steps, there are still some noises and isolated points in the resulting images, so we seek to obtain more accurate region of interests candidates of traffic signs by further morphological filtering and MSER processing.

Morphological filtering are done using the opening function of openCV which act like erosion and dilatation to get rid of the isolated points without damaging contours.

5.4. Region detection using MSER Algorithm

The MSER Algorithm follows these steps:

- Define an inscreasing sequence of thresholds $(t_j)_{j=1}^{k}$, each $t_j \in S = \{0, \ldots, 255\}$ and $\forall j \in \{0, \ldots, k-1\}, t_j < t_{j+1}$
- For each threshold t_j, we define the thresholded image:

$$\forall \mathbf{p} \in DI_{t_j}(\mathbf{p}) = \begin{cases} 255 & \text{if } I(\mathbf{p}) > t_j \\ 0 & \text{if } I(\mathbf{p}) \leq t_j \end{cases} \tag{6}$$

- For each thresholded image I_{t_j}, we determine all the maximal intensity regions $(Q_{t_j,l})_{l=1}^{m_{t_j}}$ (for each thresholded image I_{t_j} we detect m_{t_j} maximal intensity regions). To simplify our notations, we will consider that for each threshold, the number of maximal intensity regions $m_{t_j} = m$ is constant. In fact, for a small threshold (t_j near to 0), the thresholded image will be almost everywhere white. When the threshold increases, we will progressively have an thresholded image with local maxima in white. At the end, when the threshold is near to 255, the thresholded image will be entirely black. Moreover, we can see two (or more !) maximal intensity regions that can merge.

- $\forall j \in \{0, \ldots, k\}, \forall l \in \{0, \ldots, m\}$, we define

$$q(t_j, l) = \frac{|Q_{t_j,l} \backslash Q_{t_{j-1},l}|}{|Q_{t_{j-1},l}|} \tag{7}$$

- $\forall l \in \{1, \ldots, m\}$, we find $t_{j_l^*}$ such as

$$t_{j_l^*} = \underset{t_j}{\operatorname{argmin}} \ q(t_j, l) \tag{8}$$

- We finally consider $(Q_{t_{j^*},l}, t_{j^*})_{l=1}^{m}$ which represents Maximally Stable Extremal Regions (MSERs) and their corresponding thresholds

When we have MSERs, we have our regions of interests. So we can consider the MSERs contours (points in each MSERs) and their corresponding rectangular bounding boxes.

5.5. Non-maximum suppression

Unfortunately, we can have many overlapping bounding boxes after MSER algorithm. To solve this issue, we have used Faster Non Maximum Suppression. This algorithm can be describe as follow:

- Define an overlapping threshold
- Consider for instance r bounding boxes such as their overlap ratios is larger than the overlapping threshold.
- Determine the largest bounding boxe and suppress the other $r-1$ bounding boxes
- Repeat this alogithm until it remains bounding boxes that do not overlap or overlap with an opverlap ratio smaller than the overlapping threshold

At this stage, we have bounding boxes and their corresponding MSERs contours.

5.6. Convex Hull for MSERs contours

Our goal is to detect Traffic Signs. As explain previously, Traffic Signs have a particular shape (circle, square, etc...). However, contours detected by MSER algorithm can have ”some noise” at the border. So, we decided to consider the convex hull of each contours as our new contours. This step is crucial for the next step (Statistical Filering).

5.7. Statistical Filtering

We then discriminated these candidates for regions of interest based using their symmetry. Indeed, all the traffic signs present on the images of our dataset are circular, triangular or square, which are symmetrical shapes. Thus, by testing the symmetry of these regions, it will be possible for us to effectively restrict the number of candidates.

To this end, as detailed in [10], we first isolated each candidate using the bounding boxes obtained from step 3, and the readjusted contours from step 4. Then, we projected the shape onto the columns of the image by calculating the number of pixels per column. We then performed a processing on the vector C using a Minkowski distance fit, which is defined for two structuring elements A and B by $A \ominus B = [x | B \oplus \{x\} \subset A]$ [7]. Here, we will define our features vector d by the following relation: $\forall k \; 1 \leq k \leq \frac{N}{2}$,

$$d_k = min(|C_{\frac{N}{2}} - C_{N-\frac{N}{2}}|, |C_{\frac{N}{2}} - C_{N+1-\frac{N}{2}}|, |C_{\frac{N}{2}} - C_{N+2-\frac{N}{2}}|) \tag{9}$$

Indeed, a first approach could be to compare columns C_i and C_{N+1-i}, and consider that the shape is symmetrical if the difference is null, or less than a certain ϵ. However, since the contours obtained previously are not perfect, considering also the closest neighbors of the column C_{N+1-i} (before and after) is a more robust approach to the presence of offsets.

The following statistic T is considered:

$$T = \sqrt{n} \frac{-\mu_0}{S} \tag{10}$$

with $\bar{d}$ the empirical mean of the vector d, and S^2 its empirical variance which are unbiased estimators of μ and σ^2 respectively. We will then use the following statistical test to assume or not the symmetry of a region of interest :

$$\begin{cases} H_0 : \mu \leq \mu_0 \\ H_1 : \mu > \mu_0 \end{cases} \tag{11}$$

We will then distinguish two cases according to the length $n = \frac{N}{2}$ of the vector $d = [d_1 ... d_n]$: if $n < 45$, we will consider that T follows a Student's distribution with n-1 degrees of freedom. We will then reject the null hypothesis if T is greater than the quantile of order $1 - \alpha$ of the Student's distribution with n-1 degrees of freedom, if $n \geq 45$, we will consider that T follows a Gaussian distribution and we will reject the null hypothesis if T is greater than the quantile of order $1 - \alpha$ of a Gaussian distribution.

For the next step, we'll keep an α of 5%, and we will change the μ_0 parameter. Indeed, we will never be able to have a perfect symmetry with $\mu = 0$, so we can leave a certain tolerance that we will optimize later. These operations have been gathered in the symmetry function which takes as arguments an image, the bounding boxes and contours of the previous algorithms, the two parameters μ_0 and α and returns the bounding boxes of symmetric regions.

5.8. Filtering bounding boxes using geometrical hypothesis

Once we have detected the symmetry of the regions, we may still have some remaining outliers that are not traffic signs. To eliminate them, we can introduce a number of constraints on the height h and width w of the regions. Indeed, we know that on our dataset, the panels have dimensions between 16×16 and 120×120. Thus, Xu et al. have defined the following constraints which we have used in a first time :

Geometric constraints	Minimum(pixel)	Maximum(pixel)
Width	12	130
Height	12	130
Height/Width	0.8	1.2
Area pixels	144	16900

Figure 2. Constraints used in step 6

These operations are performed in function geometry which takes as argument the bounding boxes delimiting symmetrical regions, and the set of constraints previously listed as vector $[value_{min}, value_{max}$ and returns the bounding boxes satisfying these constraints.

5.9. Summery of our method

We sum up our methodology in these steps:

- **step 1** : Transfom RGB images into Normalized Red/Blue images according to equation
- **step 2** : Make Adaptive Thresholding and background brightness attenuation
- **step 3** : Extract candidates for regions of interest using filtering operations and the Maximally Stable Extremal Regions (MSER) algorithm.
- **step 4**: Use convex operations to improve the quality of the contours
- **step 5** : Analysis of candidate symmetry using statistical features to determine regions of interest
- **step 6** : Use of geometrical constraints to eliminate possible outsiders and determine the traffic signs present in the scene

6. Evaluation

To assess the relevance of our approach, we used the public dataset **GTSDB** [1] for the detection of traffic signs. It consists of several traffic images with different weather conditions in Germany. An image can contain from 0 to 6 traffic signs. The signs vary in size from 16×16 to 128×128 pixels and appear in different perspectives and brightness. This data set is divided into 900 images for training with a ground truth of the different traffic signs as bounding boxes coordinates.

We will evaluate our work using two indicators:

- the Jaccard index to measure the accuracy of sign detection in the image
- the training time of our models

We first introduce the function evaluate_solution which takes as argument the ground truth, a list containing the bounding boxes we have built with our algorithm, the images of the training set and finally the indexes of the images we are interested in. Indeed, we will see later that we had to select a part of the dataset to be able to exploit our results. This function simply consists in determining for each image the ratio intersection over union of the masks resulting from the ground truth and our prediction. It returns the average score for all the images and a list containing the score obtained for each image.

To get our predictions, we compiled all the steps into a single function get_prediction which takes as arguments the various parameters established so far, namely the delta of the MSER, the quantile and the method for thresholding, and finally μ_0 and α for the symmetry test part. As a first approach, we considered the following set which seemed to give us good results:

- delta = 2
- quantile = 0.9
- method = 'Classic'
- overlapThresh = 0.1
- μ_0 = 0.58

We got very different results from those obtained by Xu et al. While they claimed an average score of 81% over the whole dataset, with an average calculation time of 55 seconds per image, we obtain a very low score of 1% in a time of 285.9 seconds on google collab, or an average of about 32 seconds per image, which is quite close to the computing time in the article.

When we looked more closely at the scores we got, we found that many of these images actually had a score of 0, while others had a score as high as 86%. Indeed, we realized that for some images for instance, starting from the adaptive thresholding step, the traffic signs were already almost indistinguishable because of the low number of preserved points belonging to the traffic sign, and the proximity of many noisy points :

Figure 3. Example of an image before step 3 with a score of 0

Therefore, decreasing the quantile value to 0.77 to recover more points, and keeping the other parameters identical, we get the following result:

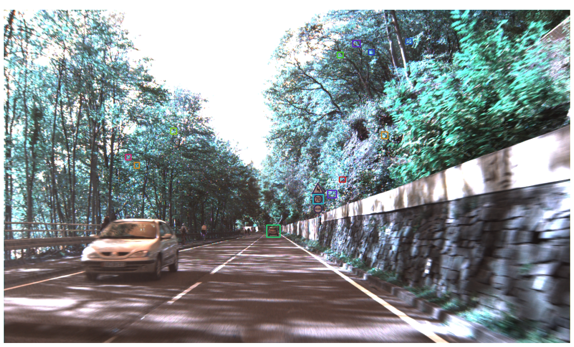

Figure 4. Example of an image with a score of 0 with another quantile value

This time, one of the traffic signs was correctly detected. This brief example aims to show that this score of 1% would not necessarily reflect a bad approach but rather a very strong dependence of the parameters on the considered images. In order to present the best results achievable with this method, we have chosen to select the 20 best images with the highest score with the previously performed evaluation. We then proceeded to an optimization of these parameters on these 20 images using Gaussian processes on the following spaces:

- dim_delta = Integer(1:5)
- dim_quantile=Real(8.8e-1:9.5e-1)
- dim_method=Categorical(['Classic','Transformed'])
- dim_overlap=Real(0.01:0.8)
- dim_mu0=Real(0.001:1.2)

After iterating over 50 steps, starting from an initial score of 47% with the previous parameters, we reached a score of 59% on these 20 images with these parameters:

- delta = 5
- quantile = 0.91
- method = 'Classic'
- overlapThresh = 0.8
- μ_0 = 1.2

Thus, this score already reflects a greater interest for this approach. Moreover, we observed that on these images, a good half of them reached good scores around 70-80-90 %, while the rest remained around 20-30-50 %. Even if we take a smaller set of images, we still observe this phenomenon of parameters image dependency. So we will have to define other methods to limit this influence. We can then observe the following result with a score of 54

Figure 5. Example of result with optimized parameters

We can see here, and even on many other images, that the drop in score is due to the presence of many small windows. We have therefore decided to change some of our geometric constraints of the last step

By increasing both h_{min} and w_{min} to 30, we see an increase in the score to 63% for those 20 images we selected, and we obtain the following result:

Figure 6. Example of result with optimized parameters with $h_{min} = w_{min} = 30$

These results can then be summarized in the following table:

Number of images	IoU	IoU with optimized parameters	idem with hmin=wmin=30
900	1%	-	-
20	47%	59%	63%

Figure 7. Results

Going back over the different steps implemented during this project, we realized that the result of the convex hull function was a set of scattered points delimiting the convex hull of the considered region, and not a set of points forming continuous contours as with MSER. This could therefore explain the poor results obtained as the study of the shapes' symmetry would then be less efficient. This could be a way to explore to improve our results. However, rapid tests carried out without the convex hull lead to similar results. The specificity of the parameters of our image algorithms thus remains the main problem.

7. Conclusion

7.1. Overview

During this project, we proposed a solution to detect traffic signs in complex environment situation. This implementation rely on several steps based mainly on the transformation of RGB images into Normalized Red/Blue images, the MSER algorithm and the analysis of candidates' symmetry.

We got mixed results, with very poor performances on average but excellent performances on a limited number of images by optimizing the parameters of our algorithms. The main problem is that there may exist an ideal set of parameters but we have to compute too many combinations to find it. Our approach also has the advantage of being innovative, but also of having very short computation times, which leaves us optimistic about the possibility of improving this work.

7.2. Future Work

Our work can be largely improved. Indeed, we propose here some improvements for Traffic Signs detection:

- Calculate the moments of the contours to have more geometrical features. It will help to filter contours (and at the same time bounding boxes) which do not represent circle, square or triangle shape
- Use other Morphological Filters (we use only the opening morphological filter)
- Train a binary classifier (an SVM or any classsical Machine Learning algoritms) on Traffic Sign masks to filter "bizarre" detected bounding boxes

References

[1] The german traffic sign detection benchmark. `http://benchmark.ini.rub.de/?section=gtsrb&subsection=news`. author: Institut Fur Neuroinformatik, year : 2013.

[2] K. D. e. a. D. Deguchi, M. Shirasuna. Intelligent traffic sign detector: Adaptive learning based on online gathering of training samples. 2011.

[3] A. H. e. a. F. Moutarde, A. Bargeton. Robust on-vehicle real-time visual detection of american and european speed limit signs, with a modular traffic signs recognition system. 2007.

[4] W. W. e. a. H.H. Chiang, Y.L. Chen. Road speed sign recognition using edge-voting principle and learning vector quantization network. 2010.

[5] E. H. e. a. I.M. Creusen, R.G.J. Wijnhoven. Color exploitation in hog-based traffic sign detection. 2010.

[6] M. M. J. Greenhalgh. Real-time detection and recognition of road traffic signs. 2012.

[7] J.Mattioli. Minkowski operations and vector spaces. *Set-Valued Anal. 3 (1)*, pages 33–50, 1995.

[8] M. S. e. a. L.E.A. De, L.E. Moreno. Road traffic sign detection and classification. pages 848–859, 1997.

[9] J. M. M. Benallal. Minkowski operations and vector spaces. 2003.

[10] S. Z. L. Z. S. P. Z. C. Xianghua Xu, Jiancheng Jin. Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry. *Future Generation Computer Systems*, 94:381–391, 2019.

[11] M. T. e. a. Y. Gu, T. Yendo. Traffic sign detection in dual-focal active camera system. page 1054–1059, 2011.

[12] X. Z. Z. Wang, Z. Tang. Reflection symmetry detection using locally affine invariant edge correspondence. pages 1297–1301, 2015.

[13] C. B. Z. Xu, J. Ren. A traffic signs' detection method of contour approximation based on concave removal. 2016.