We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Chapter

A Study on Traditional and CNN Based Computer Vision Sensors for Detection and Recognition of Road Signs with Realization for ADAS

*Vinay M. Shivanna, Kuan-Chou Chen, Bo-Xun Wu
and Jiun-In Guo*

Abstract

The aim of this chapter is to provide an overview of how road signs can be detected and recognized to aid the ADAS applications and thus enhance the safety employing digital image processing and neural network based methods. The chapter also provides a comparison of these methods.

Keywords: Advanced Driver Assistance System (ADAS), digit recognition, digital image processing, neural networks, shape detection, road signs detection, road signs recognition

1. Introduction

Increasing population elevated the demand for personal vehicles and hence evolved the advancements in the vehicular designs, engine designs, and integration of embedded electronics making the personal vehicles one of the most integrated technologies of the everyday life [1, 2]. With personal vehicles becoming ubiquitous in everyday life, there has been a rise in the associated risks. As per the data from the U.S. Census Bureau, 10.8 million vehicular accidents have been recorded in the year 2009 compared to 11.5 million accidents in the year 1990 [3] marking the reduction in accidents by 6%.

With the evolution of progressive intelligence systems popularly referred as Advanced Driver Assistance System (ADAS) comprising of lane departure warning systems, forward collision warning system, road signs (speed limit and speed regulatory) detection and recognition system, driver drowsiness and behavioral detection and alert systems, and also adoption of passive safety measures such as airbags, antilock brakes, tire pressure monitoring systems or deflation detection systems, automated parking systems, infrared night vision, pre-crash safety system and so on have not only increased the driver safety but also resulted in the reduction of associated risks as these technologies continuously monitor the driver as well as their and vehicular environment and provides timely information and warnings to the driver.

The detection and recognition of road signs is an important technology for the ADAS. Road signs are a guide to the drivers about the directions on the road, conditions of the road and serve as an essential warning under certain special road conditions. Thus, they enhance the road safety by providing the vital information. However, there might be the cases where a driver is distracted, is under stress of life, work or traffic, suffering lack of concentration or overwhelmed leading to overlook the road signs. Therefore, a system to monitor the road ahead of the vehicle, recognizing road signs and alerting about the vital conditions of the road would be an excellent assistance to the drivers. Pointedly, the road signs detection and recognition, which is the topic presented in this chapter cautions driver about the various road signs in a particular stretch of highways/roads enabling the drivers to drive within those limits, taking care of the road conditions and preventing from any over-speeding dangers.

The branch of computer science engineering that enables the machines which is the ADAS system in this case, to see, identify, interpret, and respond to the digital images and videos is termed as *Computer Vision*, abbreviated as *CV*. Until the boom of machine learning[1] techniques, CV was largely depended on traditional digital image processing (DIP)[2] methods, which are now mostly predicated on artificial neural networks (CNN)[3]. The impossible task for facilitating machines to respond to visions is achieved with the help of CV and it is intertwined with artificial intelligence[4].

The field of CV comprises of all tasks similar to biological vision systems such as seeing, i.e., visual senses, perceiving what is seen, draw detailed information in a pattern in which it can be used for further processes ultimately providing appropriate responses. In short, it is a modus operandi to instill humankind tendencies to a computer. CV finds its applications in the field of multiple disciplines aiding in simulating and automating the functions biological vision system employing sensors, computers, and various embedded platforms in assistance with numerous algorithms.

The applications of CV are enormous and broader. Of those numerous applications, using CV for detection and recognition of road signs to aid the Advanced Driver Assistance Systems (ADAS) is pivotal. This chapter focuses on road signs, also termed as traffic signs, detection and recognition using the key CV techniques.

The novelty of this chapter includes: (i) the proposed CV based method detects and recognizes the speed limit and speed regulatory signs without any templates as the templates are part of the code and not the images. (ii) the proposed CSPJacinto-SSD network enhances detection accuracy while reducing the model parameters and complexity compared to the original Jacinto-SSD.

---

[1] Machine learning (ML) is a branch of artificial intelligence (AI) and computer science, which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy [4].

[2] Digital Image Processing (DIP) refers to the use of computer algorithms to perform image processing on digital or digitized images, leading to the extraction of attributes from the processed images and to the recognition and mapping of individual objects, features or patterns [5].

[3] An Artificial Neural Network (ANN) is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates [6].

[4] Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving [7].

2. Computer vision in ADAS applications

Computer Vision (CV) is one of the crucial technologies in building the smart and advanced vehicles with autonomous driving capabilities termed as Advanced Driving Assisting System (ADAS). One of the key arena of active research of the ADAS is the road signs detection and recognition, which is a challenging task. A number of issues such as the type of camcorder, the speed of a car, noises in the image depending on speed and direction, type and intensity of light and weather conditions and sometimes the background and other objects that are similar to the signs makes it monotonous to detect and recognize the road signs. Additionally, road signs may also be damaged, faded out, tilted and partially submerged by other objects such as building signboards, trees, leading to confusion in the automated system. The process of detecting the road signs of all types is carried out using the images/videos candidates comprising of targeted road signs in case of both DIP methods and CNN methods. The road signs can be obtained from various datasets such as German Traffic Signs Dataset (GTSDB) [8], Tsinghua-Tencent 100 K [9], ImageNet dataset [10], Pascal VOC [11] to name a few. Most of these vastly used datasets may not have all the road signs in sufficient numbers captured under different lighting and weather conditions. This leads to researches to build their own datasets or rely on mechanical simulations such as CarSim [12] to build the lacking traffic signs.

This chapter discusses a low-complexity DIP algorithm and CNN based method along with the existing researches, product embodiments of these technologies followed by the algorithm design, hardware implementation and performance results of road signs detection and recognition.

2.1 Road signs detection and recognition

The process of locating the road signs from a moving vehicle followed by recognizing the exact type of road signs can be termed as '*road signs detection and recognition.*' Although there are various approaches and different algorithms, some patterns may appear similar to that of an existing body of work as in **Figure 1** that shows the basic steps employed in road signs detection and recognition flow. The process is generally divided into three parts namely, road signs detection to locate the potential candidates of road signs followed by the verification of the detected road signs' candidates from the previous stages. Finally, the recognition of traffic signs to formulate the actual information from the detected and verified signs. This task of detecting followed by recognizing road signs to aid ADAS can be achieved through both DIP and CNN based methods.

Torresen et al. [13] presents a red-colored circular speed limit signs detection method to detect and recognize the speed limit signs of Norway. Moutarde et al. [14] presents a robust visual speed limit signs detection and recognition system for American and European speed limit signs. Keller et al. [15] presents a rectangular speed limit signs detection scheme aimed at detecting and recognizing the speed limit signs in United States of America (U.S.A.). A different approach is used by Liu et al. [16] wherein the de-noising method based on the histogram of oriented gradients (HOG) is applied to Fast Radial Symmetric Transform approach to detect the circular speed-limit signs. Zumra et al. [17] and Vavilin et al. [18] both uses color segmentation followed by other digital processing methods. Lipo et al. [19] presents the method that fusions camera and LIDAR data followed by the HOG and linear SVM to classify the traffic signs.

Sebastian et al. [20] presents the evaluation of the traffic signs detection in the real-world environments. The traffic signs are detected using the Viola-Jones detector
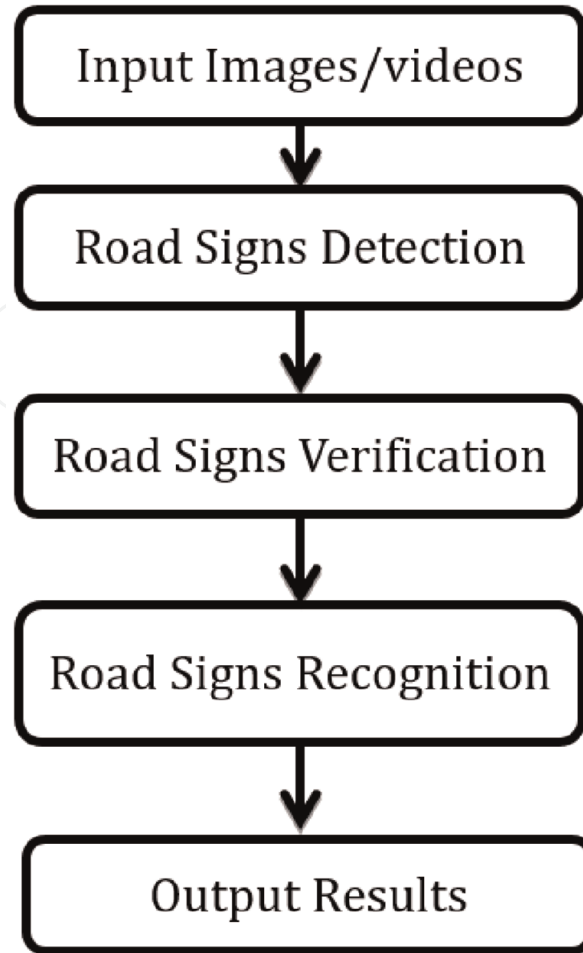
Figure 1.
Basic steps of road signs detection and recognition.

based on the Haar features and Histogram of Orientated Gradients (HOG) relied on linear classifiers. Model-based Hough-like voting methods are tested on the standard-The GTSDB. It also discusses different methods proposed by the Ming et al. [21] that uses two different, supervised modules for detection and recognition, respectively. Markus et al. [22] uses modern variants of HOG features for detection and sparse representations for classification and Gangyi et al. [23] presents the method that uses the HOG and a coarse-to-fine sliding window scheme for the detection and recognition of traffic signs, respectively.

Supreeth et al. [24] presents color and shape based detection scheme aimed at detection of red color traffic signs that are recognized using the auto associative neural networks. Nadra Ben et al. [25] presents a traffic sign detection and recognition scheme aimed at recognition and tracking of the prohibitory signs. Then Feature vector extraction along with the Support vector mechanism (SVM) is used to recognize the traffic signs and the recognized traffic signs are tracked by the optical-flow based method of Lucas-Kanade tracker [26]. Y. Chang et al. [27] adopted the modified radial symmetric transform to detect the rectangular patterns and then Haar-like feature based AdaBoost detector to reject the false positives. Abdelhamid Mammeri et al. [28] proposed an algorithm for the North American Speed limit signs detection and recognition. There are plenty of state-of-the-art researches based on different models of CNN [29–32] to detect and recognize the traffic signs including some hybrid approaches [33, 34].

2.1.1 Traditional digital image processing methods to detect and recognize road signs

The method employed to achieve certain operation on images with the aim of getting an enhanced image or extracting useful, interpretable information is termed as Image processing. It is similar to signal processing with the contraction that here input is an image and output is either an image or features affiliated with that image. In recent decades, image processing is among rapidly growing technologies. It forms the foundation for the computer vision and one of the core research area within engineering and computer science disciplines.

Fundamentally, image processing comprises of three steps namely, (i) Use of image acquisition tools to capture/import the images; (ii) Analyses and manipulation of the image; and, (iii) Output in which result can be altered image or report that is based on image analysis.

The methods used for image processing can be broadly classified in two, namely, analogue and digital image processing. Analogue image processing (AIP) refers to use of printouts and photographs to analyze via the basic interpretation employing visual techniques. On the other hand, digital image processing (DIP) techniques, as per the name, comprises of techniques that manipulate images digitally using computers. Pre-processing, enhancement, information extraction, and display are the basic, customary processes for all the data to undergo in DIP.

The process of detection and recognition of speed limit road signs [35] can be broadly classified into three stages namely, (i) speed limit signs detection, (ii) digit segmentation, (iii) digit recognition and that of detection and recognition of speed regulatory road signs [36] also into three stages such as, (i) speed regulatory signs detection, (ii) feature extraction, (iii) feature matching. **Figure 2** depicts the proposed algorithm used in detection and recognition of the road signs. The following sections discuss each step of the algorithm and the corresponding implementation specifications of the respective stages.

A. Shape Detection

The process of detecting regular and irregular polygon shapes is termed as *Shape detection*. Shape detection in this chapter refers to detecting the road signs which processes the entire frame and then focuses on selecting the potential candidates of size 32x32, 64x64 and so on comprising the common shapes, either a circle or a rectangle of the speed limit sign or a triangular signs of the speed-regulated signs using radial symmetric transform method.

The concept of radial symmetric transform [37, 38] uses the axes of radial symmetry. The normal polygenes of n-sides possess several axes of symmetry and the radial symmetric transform works based on these symmetric axes.

The voting process is based on the gradient of each pixel [39]. The direction of gradient generates a vote. The vote generated from each pixel follows the symmetric axes resulting in the highest votes at the center of the respective symmetric axes. **Figure 3** shows the radial symmetry for common polygenes.

Fundamentally, the Sobel operator [40] is applied to calculate the gradient of each pixel using a Sobel mask. Sobel operator generates the gradient of the intensities in the vector forms with the horizontal gradient denoted by G_x and vertical gradient denoted by G_y by convolving corresponding Sobel masks defining the direction of the gradients for each pixel. Besides, in order to eliminate the noises of small magnitudes, the
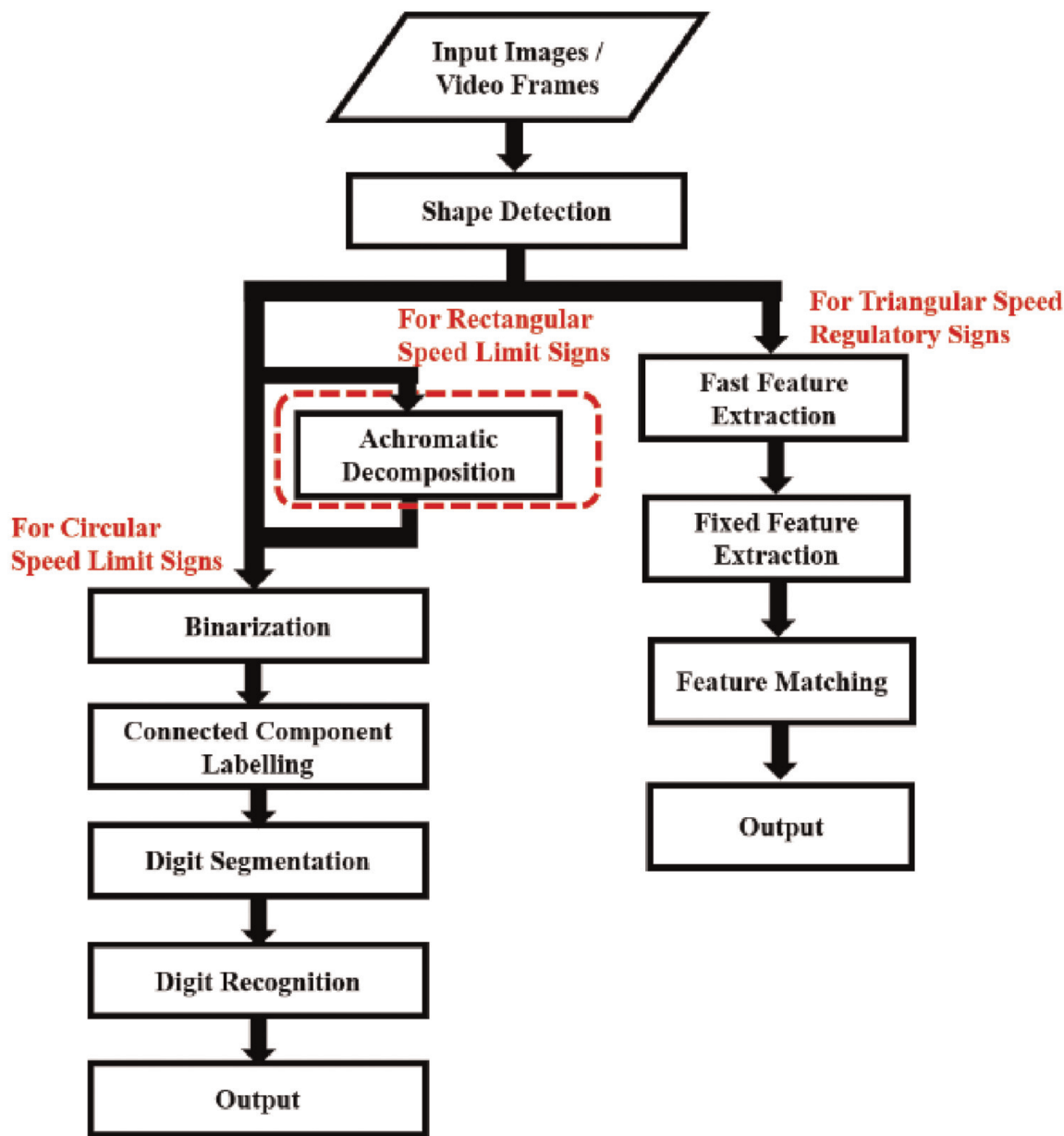
Figure 2.
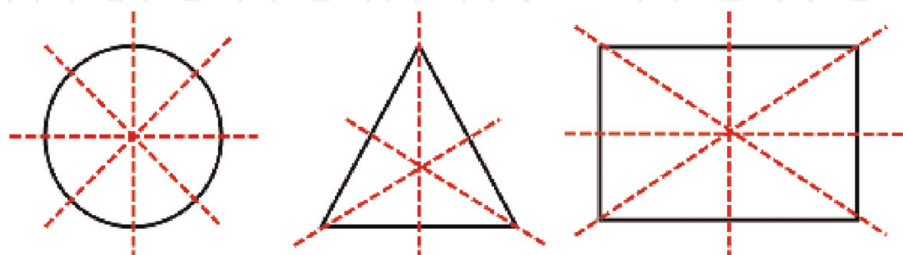*Flow chart of the proposed algorithm used to detect speed limit and speed regulatory signs.*

Figure 3.
Radial symmetry of common polygenes.

threshold for the absolute values G_{abs} is set for horizontal and vertical gradient given by Eq. (1)

$$G_{abs} = |G_x| + |G_y| \tag{1}$$

Once the horizontal gradient G_x and the vertical gradient G_y is obtained and the noise is eliminated, the radial symmetric transform can be processed based on the calculated gradients. **Figure 4** shows the results of the horizontal and vertical gradients.

     i. Rectangular Radial Symmetric Transform

The voting process in the rectangular radial symmetric transform phase is based on the gradient generated from the Sobel operator [13, 27]. Each selected pixel with its absolute magnitude G_{abs} greater than a small threshold is denoted as p, and the gradient vector is denoted as $g(p)$. The direction of $g(p)$ can be formulated with the horizontal gradient G_x and the vertical gradient G_y into an angle using Eq. (2).

$$g(p) = \tan^{-1} \frac{G_y}{G_x} \qquad (2)$$

For each considered pixel p, the votes along with the known width W and height H are divided into two categories- horizontal vote and the vertical vote. The direction of gradient $g(p)$ for each pixel is adopted to implement these two categories. The magnitude ranges and the ratio between G_x and G_y is used to verify the horizontal and the vertical votes with respect to the higher threshold and lower threshold values.

    a. If $G_x >$ higher threshold and $G_y <$ lower threshold, vote is regarded as horizontal gradients.

    b. If $G_x <$ lower threshold and $G_y >$ higher threshold, vote is regarded as vertical gradients.

Here, the values of higher threshold and lower threshold are experimentally chosen based on the size of the Sobel mask. In the case of 3x3 Sobel mask, the higher threshold is set in the range of 45–55, and the lower threshold is between 15 and 25. In case of the nighttime scenarios, both the thresholds are lowered to half of their original values and constrains are set for the ratio of horizontal and vertical gradients.

Each pixel contributes a positive vote and a negative vote. A voting line is then generated by each pixel with both positive and negative votes. The positive votes indicate the probable center of the speed limit sign while the negative votes indicate
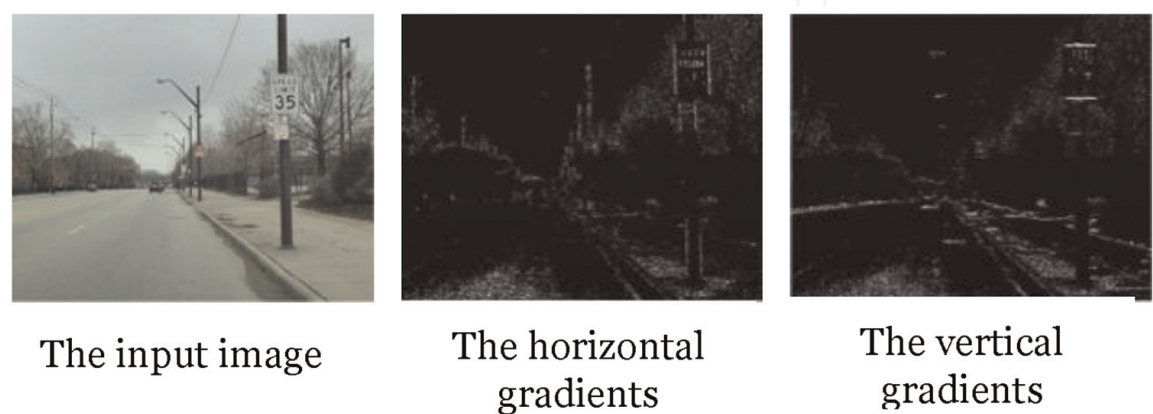

The input image   The horizontal   The vertical
                     gradients   gradients

Figure 4.
The results of the horizontal and vertical gradients.

the non-existence of speed limit signs. The positive horizontal votes $V_{horizontal+}$ and negative horizontal vote $V_{horizontal-}$ votes are formulated as in Eqs. (3) and (4). $L_{horizontal}\ (p, m)$ describes a line of pixels ahead and behind each pixel p at a distance W given by Eq. (5).

$$V_{horizontal+}\left\{ L_{horizontal}(p,m) | m \in \left[-\frac{W}{2}, \frac{W}{2} \right] \right\} \tag{3}$$

$$V_{horrizontal-}\left\{ \begin{array}{l} L_{horrizontal}(p,m)|m \in \left[-W, -\frac{W}{2} \right) \\ \cup \left(\frac{W}{2}, W \right] \end{array} \right\} \tag{4}$$

$$L_{horrizontal}(p,m)round(m * \overline{g}(p) + W * g(p)) \tag{5}$$

where $\overline{g}(p)$ is a unit vector perpendicular to $g(p)$. **Figure 5(a)** represents the process of horizontal voting. Similarly, the positive and negative vertical votes are formulated as in the Eqs. (6) and (7). $L_{vertical}\ (p, m)$ describes a line of pixels ahead and behind each pixel p at a distance W given by Eq. (8), and as shown in **Figure 5(b)** where $\overline{g}(p)$ is a unit vector perpendicular to $g(p)$.

$$V_{vertical}\left\{ L_{vertical}(p,m) | m\epsilon \left[-\frac{H}{2}, \frac{H}{2} \right] \right\} \tag{6}$$

$$V_{vertical}\left\{ L_{vertical}(p,m) | m\epsilon \left[-H, -\frac{H}{2} \right) \cup \left(\frac{H}{2}, H \right] \right\} \tag{7}$$

$$L_{vertical}(p,m) = p + round(m * \overline{g}(p) + W * g(p)) \tag{8}$$

After this voting process, the centers of the sign candidates will receive higher votes. The voting image is initially initialized to zero, and then it goes on accumulating both the positive and the negative votes. **Figure 6** shows the result for rectangular signs after the voting process.
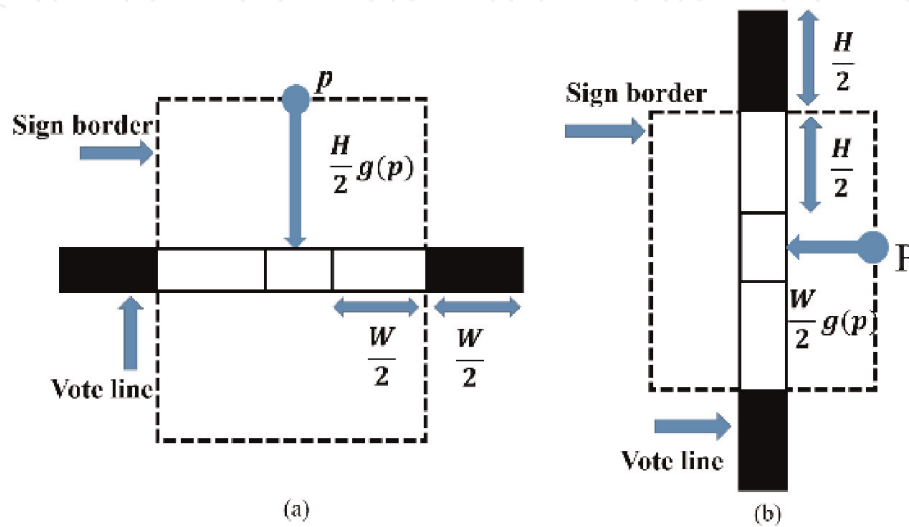

(a) (b)

Figure 5.
(a) The voting line corresponding to the horizontal voting. (b) the voting line corresponding to the vertical voting.

Figure 6.
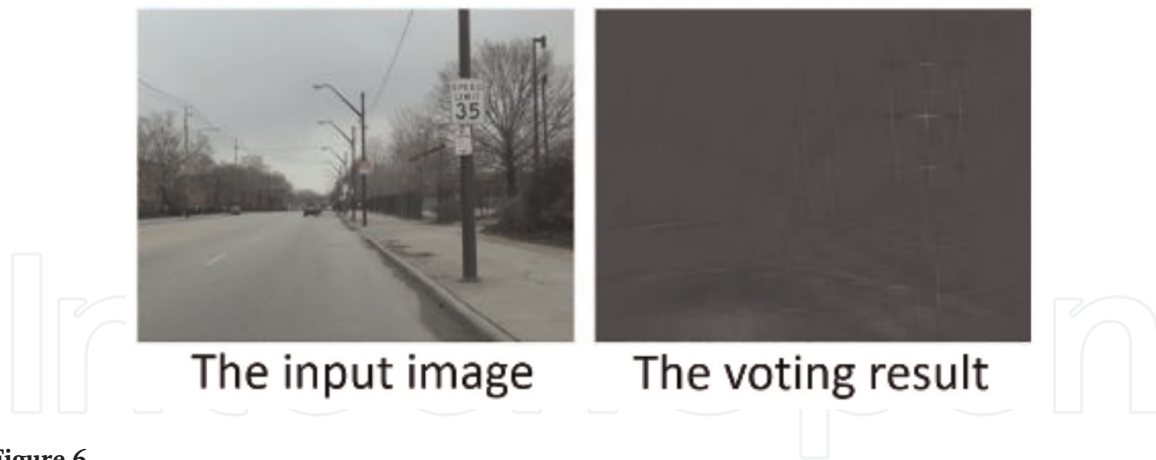*The result after the horizontal voting process.*

ii. Circular Radial Symmetric Transform

The detection of circular speed limit signs using radial symmetric transform method is similar to that of the detection of the rectangular signs with a difference that the circular radial symmetric transform need not to be divided into two parts as horizontal votes and vertical votes. It is entirely based on the direction of gradient for each pixel $g(p)$ and each considered pixel contributes only positive votes V_+ as in Eq. (9).

$$V_+ = p + \text{round} \left(R * g(p) \right) \qquad (9)$$

Figure 7(a) illustrates the voting process for the circular sign detection and **Figure 7(b)** shows the result for circular signs after the voting process.

iii. Triangular Radial Symmetric Transform

The voting process of the triangular shape detection is also based on the gradient of each pixel [39]. The vote generated from each pixel follows the rule of the proposed triangle detection algorithm shown in **Figure 8**. It comprises of: (i) Sobel operator is applied to calculate gradient for each pixel. Consequently, we calculate the horizontal and vertical gradients by convolving the corresponding Sobel masks. Each selected pixel is represented with its absolute magnitude, and the gradient vector is denoted as $g(p)$. The direction of $g(p)$ can be formulated with the horizontal gradient G_x and the vertical gradient G_y into an angle as shown in Eq. (10). Only 180 degrees of gradient is used in this algorithm followed by the morphological erosion to eliminate noises.
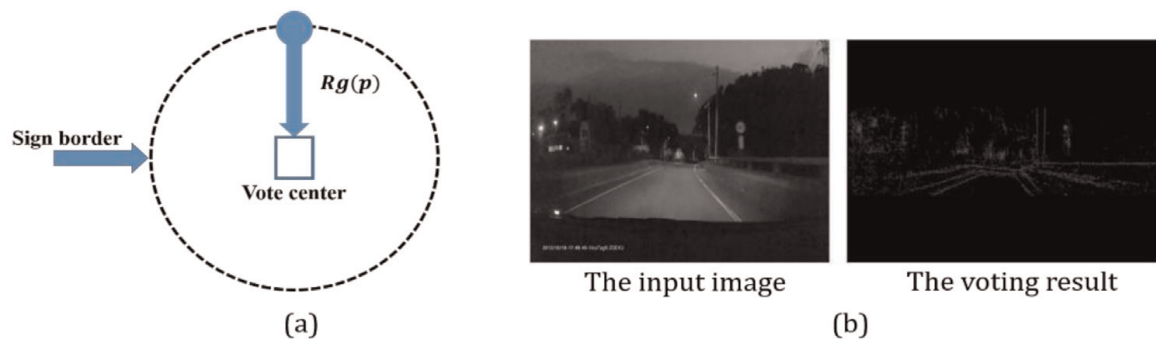

Figure 7.
(a) The vote center corresponding to (9); (b) the result of the circular voting result.
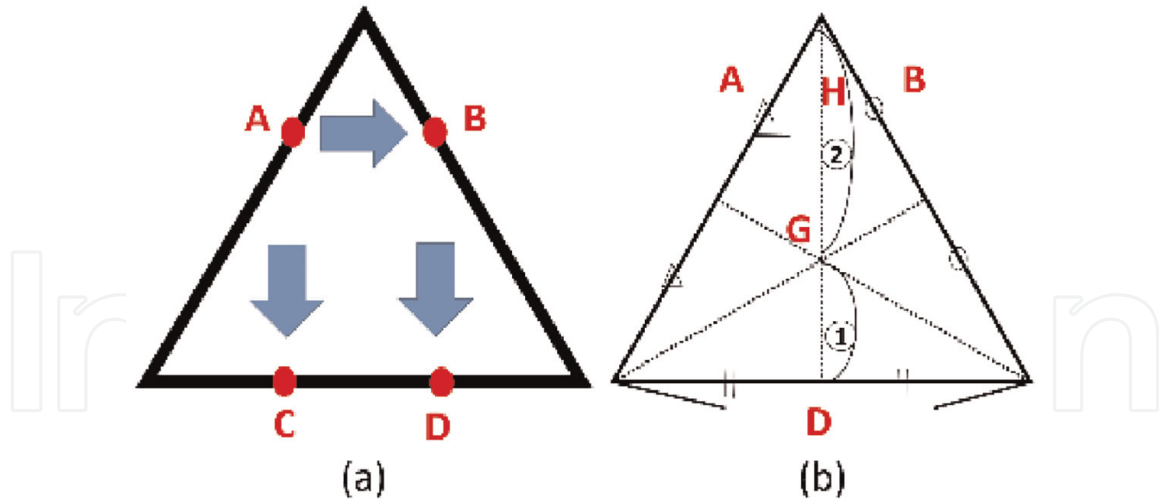
9

Figure 8.
(a) Illustration of triangle detection algorithm; (b) illustration of voting for the center.

Morphological erosion is applied to eliminate the noise. For a pixel $p(x,y,f(x,y))$ and a structure element $b(i,j)$, the formula of erosion is as in Eq. (11).

$$g(p) = tan^{-1}\frac{G_y}{G_x} \tag{10}$$

$$(f \ominus b)(x,y) = e(x,y) = \min_{(i,j)} (f(x+i,y+j) - b(i,j)) \tag{11}$$

Then the proposed algorithm exploits the nature of triangle for the detection as in **Figure 9(a)**. We look for the points having a gradient of 30 degree, defined as point A. Once the point A is obtained, we search for the points that have a gradient of 150 degrees on the same row as that of point A, defined as point B. The last step is to find the points C and D with 90 degrees gradient on the same column with points A and B, respectively. Once all these points are determined, a vote is placed to the point G at the centroid of the triangle as in **Figure 9(b)**.

In order to vote for the center point, the width of the detected position and the size of the target triangle is required to be calculated. The formulas shown in Eqs. (12) and (13), where $Center_x$ is the x-coordinate of G, $Center_y$ is the y-coordinate of G, H is the height of the small triangle, and D is the size of the target triangle, as shown in **Figure 9(b)**.

$$Center_x = \frac{A_x + B_x}{2}; H = (B_x - A_x) * \frac{\sqrt{3}}{2} \tag{12}$$

$$Center_y = \begin{cases} A_y + D * \dfrac{\sqrt{3}}{3} - H, H < D * \dfrac{\sqrt{3}}{3} \\ \\ A_y + H - D * \dfrac{\sqrt{3}}{3}, H > D * \dfrac{\sqrt{3}}{3} \end{cases} \tag{13}$$

Then the width of the detected position is needed for calculating the detected points A and B. Thus, it is easy to build a look up table to reduce the computation cost
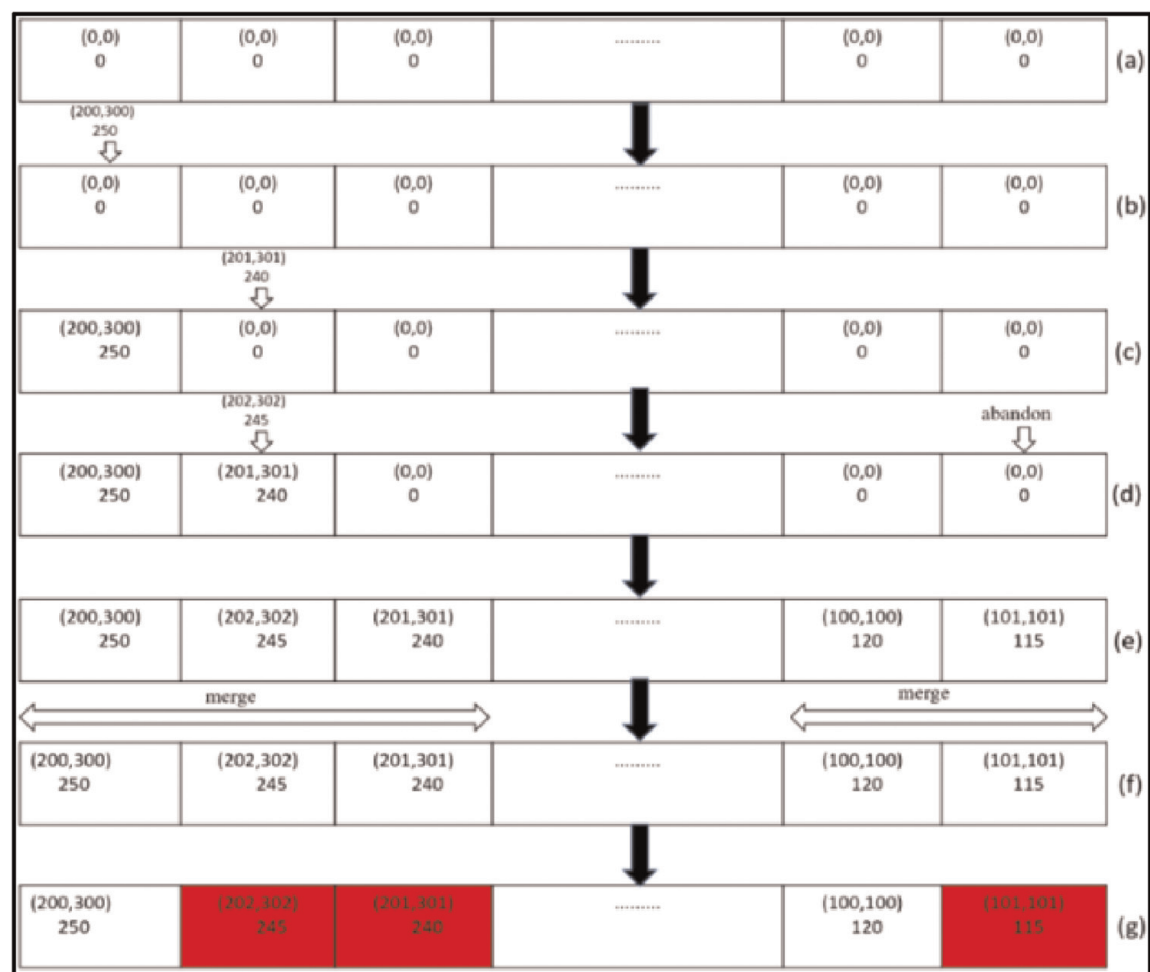
Figure 9.
The steps in detail of the sign candidate extraction (a) initialization of elements in the buffer to zero. (b), (c) insert the sign candidates based on the vote value. (d) if the current sign candidates own the greater vote value than any element in the buffer, firstly shift the element and the other elements in the wake of the right for the one-element and abandon the last element, and update the value to the element. (e), (f) post several iterations, the buffer is full of the sign candidates, and merge the cluster, only leaving the element with the greatest vote value. (g) the elements in red are merged.

of the voting process. The pixels with higher votes are judged to be in and near the center of the triangular road sign candidates. In order to reduce the computation cost, candidates that are close to each other will be merged into one candidate. The new coordinates of the candidate is the weighted arithmetic mean calculated using the coordinates of the merged candidates weighted by their vote thereby reducing the different candidates representing the same triangle.

B. Sign Candidates Extraction

After detecting the shape, the potential candidates of the road signs are extracted. A buffer is created to save all the potential sign candidates according to the following steps:

a. Initially, all the vote values in the buffer are set to zero.

b. For each vote in the input image, if the vote is greater than an experimentally set threshold, the pixel is considered. The considered vote is arranged into the

buffer based on its vote value, ensuring the buffer is in a decreasing order. Every time this buffer is sorted in a decreasing order to ensure that, the pixels with greater values are in a prior order.

The values thus generated by the votes of the sign candidates result in a cluster of candidates in a small region. To overcome this challenge, the distance and search is defined from the prior order in the buffer by setting a small distance threshold to merge the cluster of sign candidates using non-maximum suppression as per Eq. (14). where x and y are the coordinates of the current considered sign candidates, and x_i and y_i are the sign candidates of the threshold candidates. **Figure 9** illustrates the details of the sign candidates' extraction along with the results of merging the cluster of sign candidates and the results of merging the clusters of the sign candidates.

$$Distance = |x-x_i| + |y - y_i| \qquad (14)$$

C. Achromatic Decomposition

The key feature of the rectangular speed limit road signs in USA is that all the common speed limit signs are in gray-scale as in **Figure 10(a-d)**. There also exits advisory speed limit signs on the freeway exits as in **Figure 10(e-f)**. In order to detect the actual speed limit signs as in **Figure 11**, the achromatic gray scale color of the signs
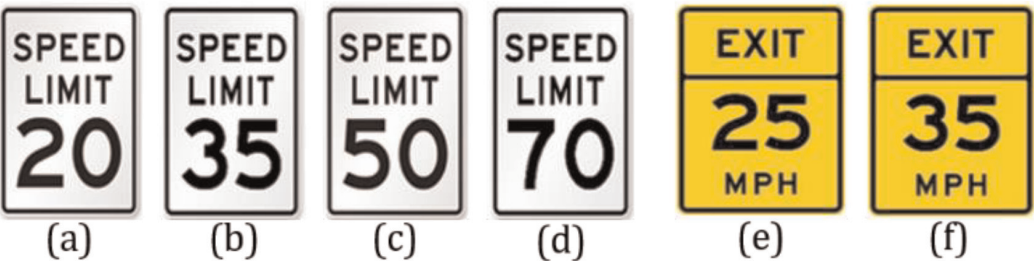

Figure 10.
(a-d) the rectangular speed limits road signs in USA. (e-f) the advisory speed signs on freeway exits.
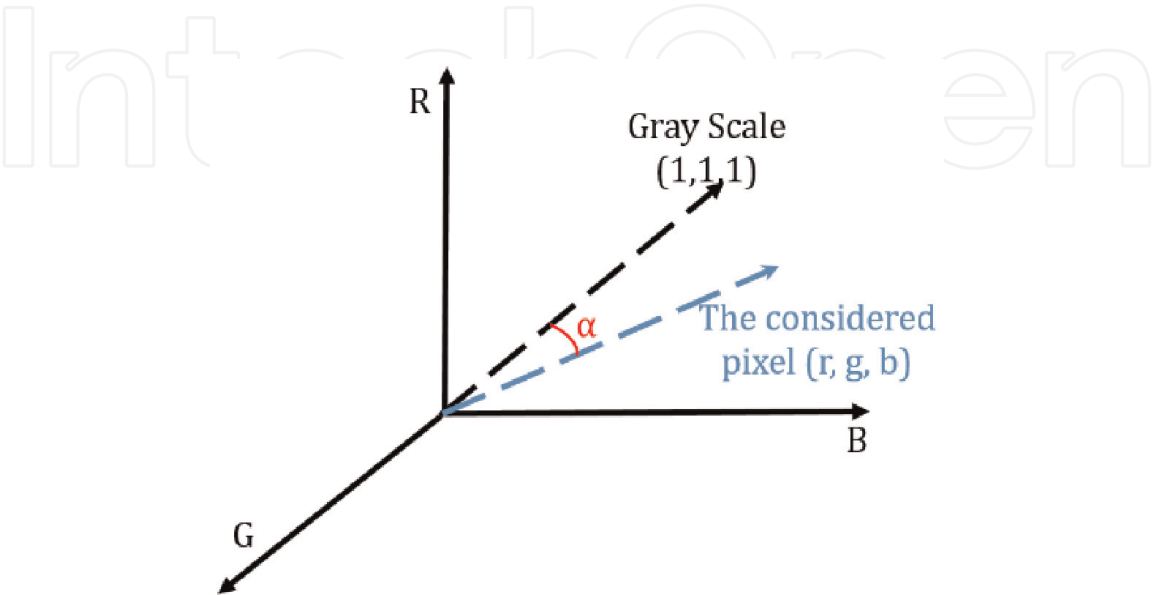

Figure 11.
The schematic of the RGB model and the angle α.

is extracted by the achromatic decomposition whereas the non-gray scale advisory speed signs are ignored from the further consideration.

The vector of gray scale is along (1,1,1) in RGB color space and the inner product is between (1,1,1) and the each considered pixel checks an angle α between these two vectors to apply the decomposition in RGB domain [41] as illustrated in **Figure 10(a-d)**.

Each of the considered pixel is in the vector form of (r, g, b). The cosine function of α, which is equal to the inner product [26] is shown in Eq. (15).

$$Cos\alpha = (1, 1, 1) \cdot (r, g, b) = \frac{r + g + b}{\sqrt{3} \times \sqrt{r^2 + g^2 + b^2}} \tag{15}$$

For the implementation in our proposed, a mere value of cos^2 is considered. If the value is near to one, α is near to zero, which implies the considered pixel is in gray-scale and is taken in account for the further steps. **Figure 12** shows the results of the achromatic decomposition where the speed warning signs of non-gray scale found on the freeway exits are not acknowledged.

D. Binarization

In the proposed system, the Otsu threshold method is used for the daylight binarization while the adaptive threshold method during the nighttime. **Figure 13** illustrates these proposed steps.

To differentiate between daylight and the night-light, the ROI is set to the cent er part of the frame choosing the width ROI_w and height ROI_h as in Eqs. (16) and (17) where the sky often lies. The noise with extremely high and low pixel values are filtered out and then in the remaining 75% of the pixels, the average of pixel values is calculated to judge if it is a day-light or a night condition. **Figure 14** shows the schematic of the day and nighttime judgment.

$$ROI_h = \frac{Height\ of\ the\ frame}{6} \tag{16}$$

$$ROI = \frac{2}{3} Width\ of\ the\ frame; Excluding\ \frac{Width\ of\ the\ frame}{6} on\ either\ ends \tag{17}$$

The Otsu method [42] can automatically decide the best threshold to binarize well in daytime, but at night, the chosen threshold causes the breakage of the sign digit.

Figure 12.
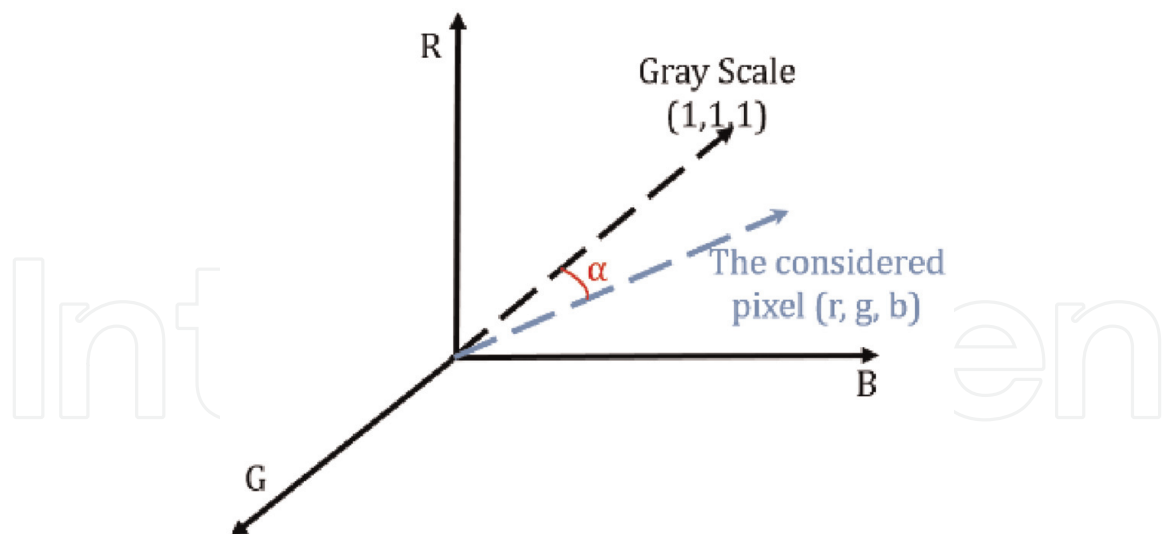*The results of the achromatic decomposition.*

Figure 13.
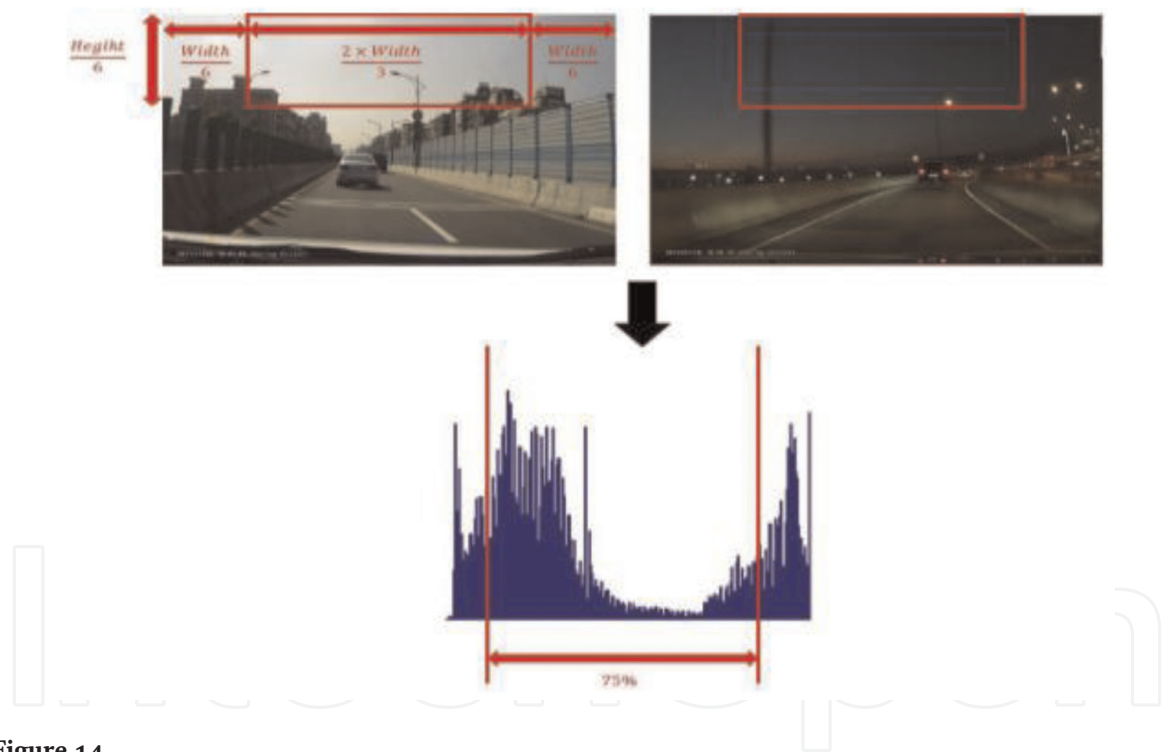*The proposed steps of binarization.*

Figure 14.
The schematic of the day and night judgment.

In Otsu method, the best threshold that can divide the content into two groups and minimize the sum of variances in each group is found by an iterative process. **Figure 15** shows the schematic steps of Otsu method. On the other hand, the adaptive threshold is more sensitive. It divides the signs into several sub-blocks, mean of each block is calculated, and then the threshold for respective sub-blocks is computed based on the means in each sub-block. The corresponding results of the adaptive threshold is as shown in **Figure 16**.

Therefore, we chose Otsu threshold to automatically select the best-fit threshold in automatically daytime and adaptive threshold at night to handle the low contrast environment.
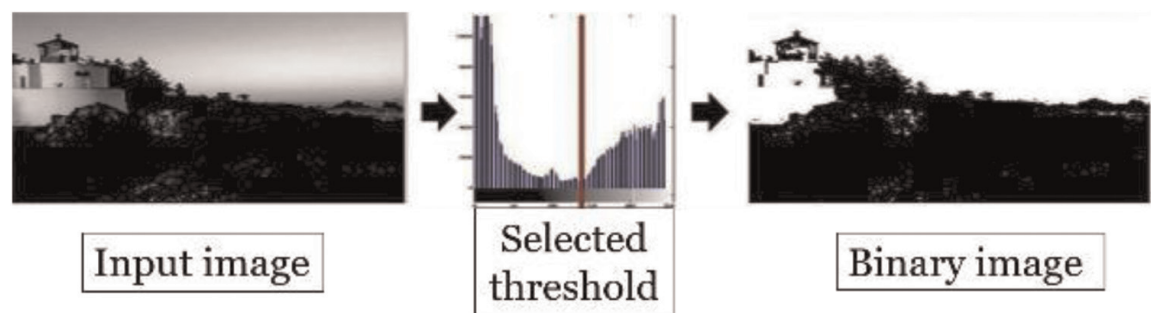
Figure 15.
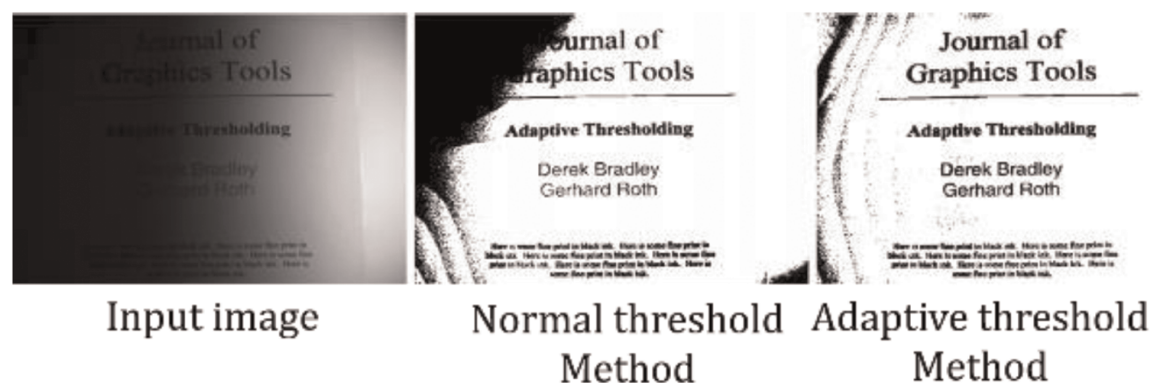*The schematic steps of Otsu method.*

Figure 16.
Different thresholding results.

For acceleration, this paper adopts the integral image [36] in which each pixel is compared to an average of the surrounding pixels. An approximate moving average of the last *s*-pixels seen is specifically calculated while traversing the image. If the value of the current pixel is lower than the average then it is set to black, otherwise it is set to white. In the proposed algorithm, it is considered to be stored at each location, $I(x, y)$, the sum of all $f(x, y)$ terms to the left and above the pixel (x, y). This is accomplished in linear time using Eq. (18) for each pixel. Once the integral image is first calculated, the sum of the function for any rectangle with the upper left corner (x_1, y_1), and lower right corner (x_2, y_2) can be computed in constant time using Eq. (19). The schematic of Eq. (13) can be illustrated with **Figure 17** and Eq. (19) can be modified into Eq. (20). Finally, the mean of each sub-block can be calculated and then
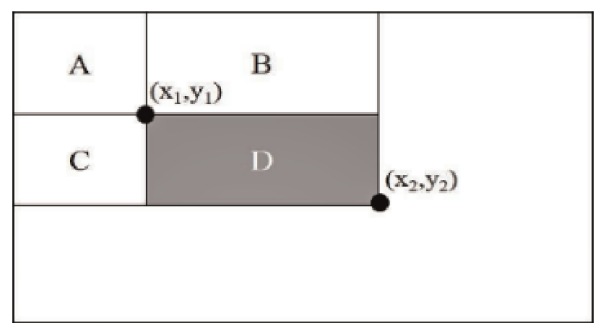
Figure 17.
The schematic figure of Eq. (18).

each pixel in the sub-block which in terms of $A(x, y)$ can be binarized with Eq. (21) where $\propto$ is a scalar based on the contrast under different conditions.

$$I(x,y) = f(x,y) + I((x-1),y) + I(x,(y-1)) - I((x-1),(y-1)) \tag{18}$$

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x,y) = I(x_2,y_2) - I(x_2,y_1-1) - I(x_1-1,y_2) + I(x_1-1,y_1-1) \tag{19}$$

$$D = (A+B+C+D) - (A+B) - (A+C) + A \tag{20}$$

$$A(x,y) = \begin{cases} 255, & \text{if } I(x,y) > T_{avg} \times \propto \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

E. Connected Component Labelling (CCL)

Connected component labelling (CCL) labels the object inside the sign candidates with the height, width, area and coordinate information. The CCL algorithm [28] is divided into two processing passes namely first pass and the second pass. Two different actions are taken in these passes if the pixel iterated is not the background.

The steps of connected component labeling are illustrated in **Figure 18**. In this case, the equivalent labels are (1, 2), (3, 7) and (4, 6).

F. Digit Segmentation

The purpose for digit segmentation [43] is to extract the digit from the binarized image. In the process of rectangular speed limit signs detection, the signboards have the characters reading "SPEED LIMIT" alongside the digits. As a result, it is necessary to set constrains on size of the digit candidates as per Eqs. (22) and (23). Similarly, the constraints on the size of the digit candidates in circular speed limit signs are as per Eqs. (24) and (25).

$$0.15 \times W \leq \text{Digit width} \leq 0.5 \times W \tag{22}$$

$$0.15 \times H \leq \text{Digit height} \leq 0.5 \times H \tag{23}$$

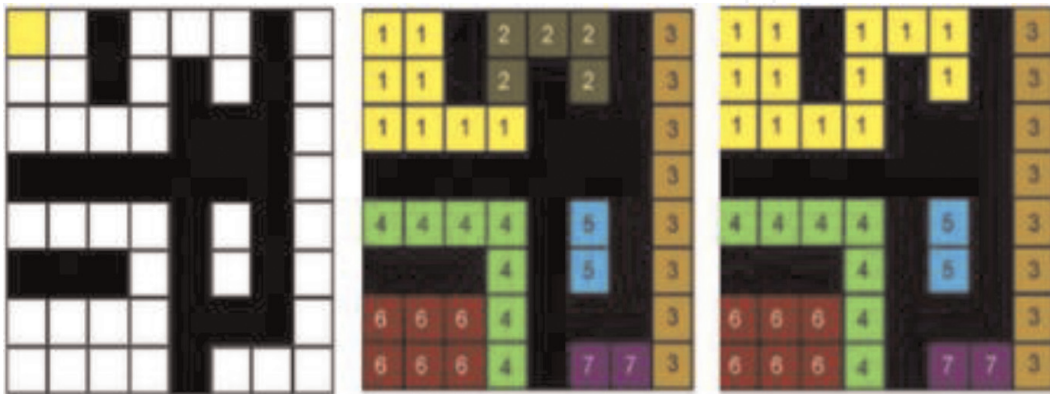$$0.125 \times R \leq \text{Digit width} \leq R \tag{24}$$

Figure 18.
The steps for connected-component labeling, (a) processing initialization, (b) the result after the first pass, (c) the result after the second pass.

$$0.5 \text{ x R} \leq \text{Digit height} \leq 1.5 \text{ x H} \qquad (25)$$

Considering the fact that rectangular speed limit signs consist two-digits alongside the characters, it must be ensured that the selected candidates are of digits of speed limit sign and not the characters. The pairing rule of sizes and positions proposed in this paper are as follows:

a. The areas of the digit candidates should be similar.

b. The positons of the digit candidates should be closely packed.

c. The density of the pixels inside the digit candidates should be similar.

Whereas in the circular speed limit detection that are of both 2-digits and 3-digits, a loose constrain is adopted as it has only digits inside the circular speed limit signs and no characters. The pairing steps are similar to those followed in the rectangular speed limit signs. **Figures 19** and **20** shows the segmentation results of the rectangular and circular speed limit signs, respectively.

There exists a critical challenge in binarization process, as the digits may appear connected to each other. To overcome this challenge, a two-pass segmentation process is proposed in this paper. Digit segmentation, similar to the previous segmentation
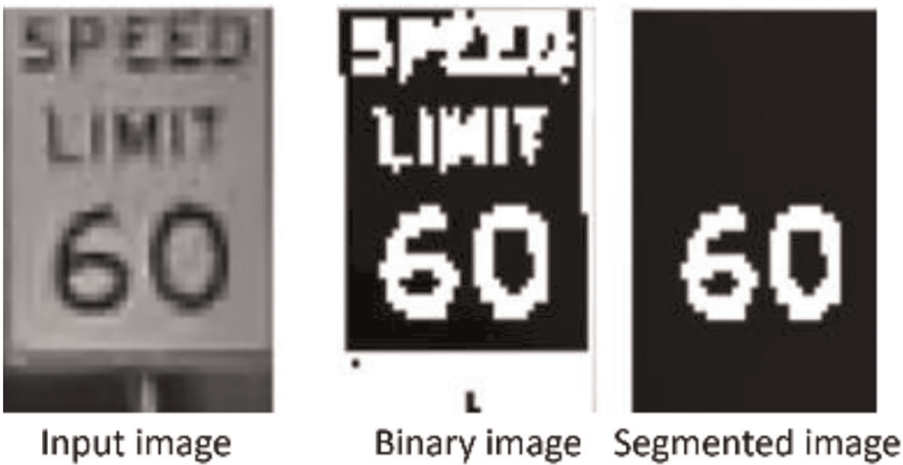

Figure 19.
The example of digit segmentation results of rectangular speed limit signs
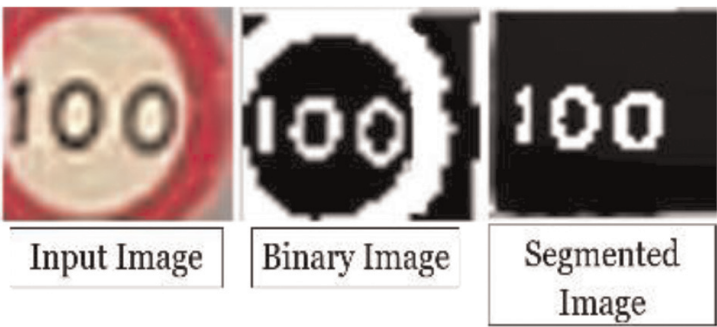

Figure 20.
The example of digit segmentation results of circular speed limit signs.

process, is applied. If large components are detected, then the second pass of the segmentation is applied.

The horizontal pixel projection is applied in the second pass segmentation. During this projection, the total number of pixels in each column is accumulated choosing the segment line based on the horizontal projection. **Figure 21** shows the example of horizontal projection and the result of the two-pass segment step.

G. Digit Recognition

In the digit recognition phase, the extracted digits are compared with the built-in templates and three probable digits with least matching difference are selected based on Sum of Absolute Difference (SAD) [44]. After which, the blob and breach features of the digits are applied to verify the final digits [45–47]. **Figure 22** depicts the proposed steps for digit recognition.

After selecting the three probable digit candidates, the blob feature is employed to verify the digit. Here the blob is defined as a close region inside the digit and is detected and gathered in several rows forming a union row. The pixel value of a union row is the union of values of the rows in that union row. For each union row, the number of lines in the white pixels are counted. A blob is formed only if the number of lines is of the sequence "1, 2,. … , 2, 1". This union row method yields the position and

| Input image | Binary image | 1st segment image | Horizontal projection | 2nd segment image |
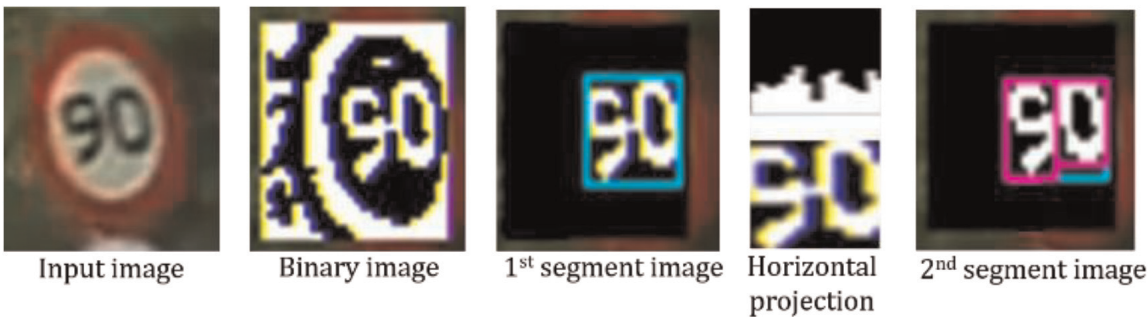
Figure 21.
Example of horizontal projection and the result of the proposed two-pass segmentation steps.

Template Matching
↓
Blob Feature Verification
↓
Breach Feature Verification
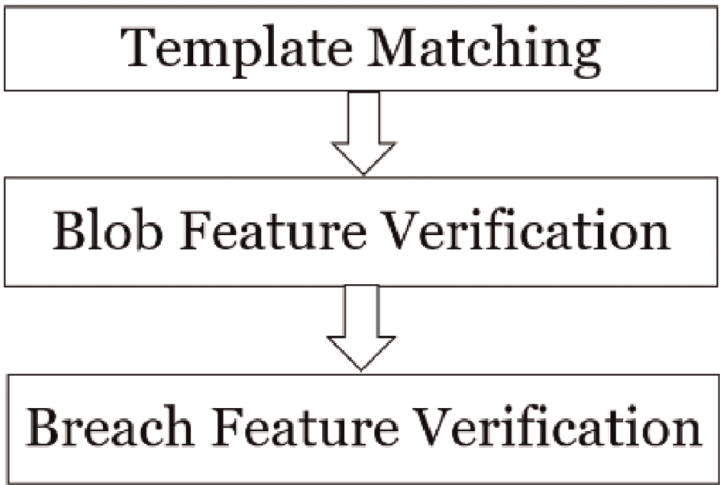
Figure 22.
The proposed steps for digit recognition.

Digit	Blob feature verification		Breach feature verification	
	Number of blobs	Position of blobs	Number of breaches	Direction of breaches
0	One	Top to Bottom	Zero	—
1	Zero	—	Zero	None
2	Zero	—	Two	Top Left & Bottom Right
3	Zero	—	One	Left
4	One	Top Left	Zero	—
5	Zero	—	Two	Top Right & Bottom Left
6	One	Lower Half	One	Top Right
7	Zero	—	One	Left
8	Two	Upper & Lower Halves	Zero	—
9	One	Upper Half	One	Bottom Left

Table 1.
The blob and breach feature verification for all the digits.

the number of the blobs in the digit candidate as in **Table 1**. The exact blob feature is defined for the specific digits.

Similarly, the breach feature is also adopted to verify the digits. A breach is defined as an open region formed by a close region with a gap. The breach is detected by counting the number of pixels where the white pixel first appears from both the right and the left in each column to half the width of the digit candidates. If there is a series of pixels that are larger than half the digit height, then it is regarded as a breach. **Table 1** shows the blob and breach feature verification for the digits from 0 to 9 and **Figure 23** shows the results of the digit recognition in terms of blob and breach feature verification.

H. FAST Feature Extraction

Features from Accelerated Segment Test (FAST) [48, 49] is a high repeatability corner detector. As shown in **Figure 24**, it uses a circle of 16 pixels to classify whether or not a candidate point p is a corner. The FAST feature extraction conditions can be written as in Eq. (26) where S is a set of N contiguous pixels in the circle, I_x is intensity of x, I_p is intensity of candidate p and t is the threshold.
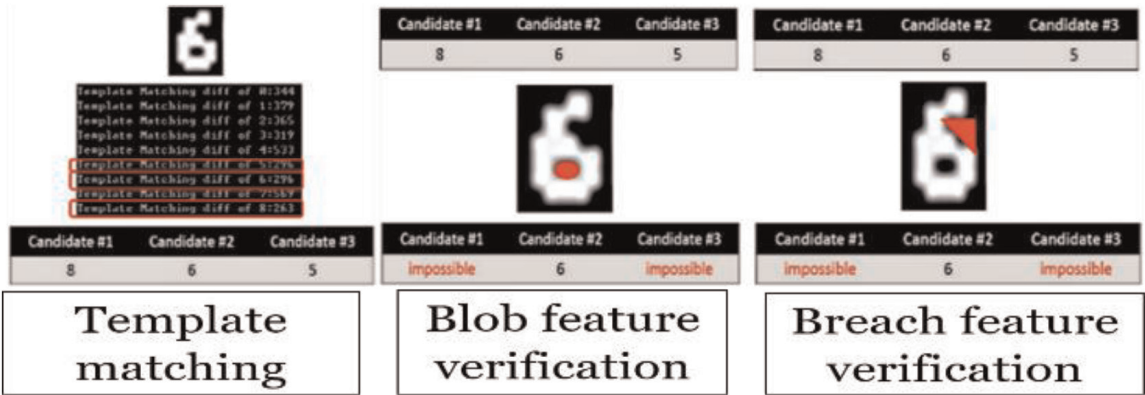

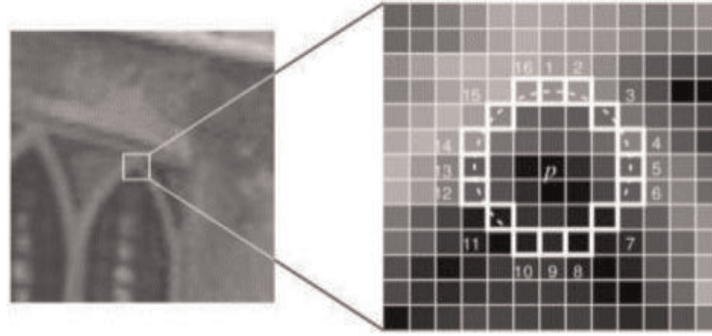**Figure 23.**
Digit recognition results.

Figure 24.
The illustration of FAST algorithm.

$$\forall x \in S, I_x > I_p + t, \text{ and } \forall x \in S, I_x < I_p - t \tag{26}$$

There are two parameters to be chosen in FAST algorithm namely, the number of contiguous pixels N and the threshold t. N is fixed as 9 in the proposed algorithm whereas to overcome the changes in the intensity inclinations as shown in **Figure 25**, the threshold t is set to be dynamic. The dynamic threshold is calculated by the image patch of the sign candidates. First, we count for pixels with intensity bigger than 128. If the number of bright pixels is between 20% and 80% of the total number of pixels in the image patch, the threshold is computed by the percentage of number of bright pixels over the total number of pixels. There are two fixed thresholds for the conditions that the number of bright pixels is lower than 20% of the total number of pixels or higher than 80% of the total number of pixels. Accordingly, the threshold dynamically updates to the number of bright pixels over the total number of pixels.

I. Fixed Feature Extraction

There are certain conditions in which the contents of road signs are too simple to be extracted using the FAST feature, as shown in **Figure 26(a-c)**. Thus, the Fixed Feature Extraction is applied to handle these road signs with good inclinations in the proposed algorithm.

Fixed Feature Extraction uses thirty fixed feature points to describe a road sign, as shown in **Figure 26(d)**. This method is similar to template matching, but it is more robust to noises as it uses descriptors to describe the small area around the feature points.

Figure 25.
Different lighting conditions of road signs.
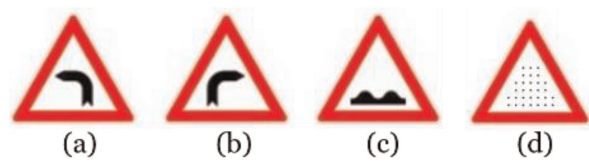
Figure 26.
(a-c) Road signs with simple contents; (d) thirty fixed feature points used in fixed feature extraction.

J. Feature Matching

The main objective of this phase is to match the features between the pre-built template and the detected sign candidates as shown in **Figure 27**. The features extracted previously are matched by their coordinates and the descriptors which are constructed to describe their respective features. Due to the fact that the proposed system is aimed at real-time applications, the construction and the matching procedure of the descriptor algorithm should be both simple and efficient at the same time.

Figure 27.
Steps followed in the feature matching.

Binary Robust Independent Elementary Features (BRIEF) [50] is a simple descriptor with good matching performance and low computation cost. In order to build a BRIEF descriptor of length n, n pairs (x_i, y_i) are chosen. X a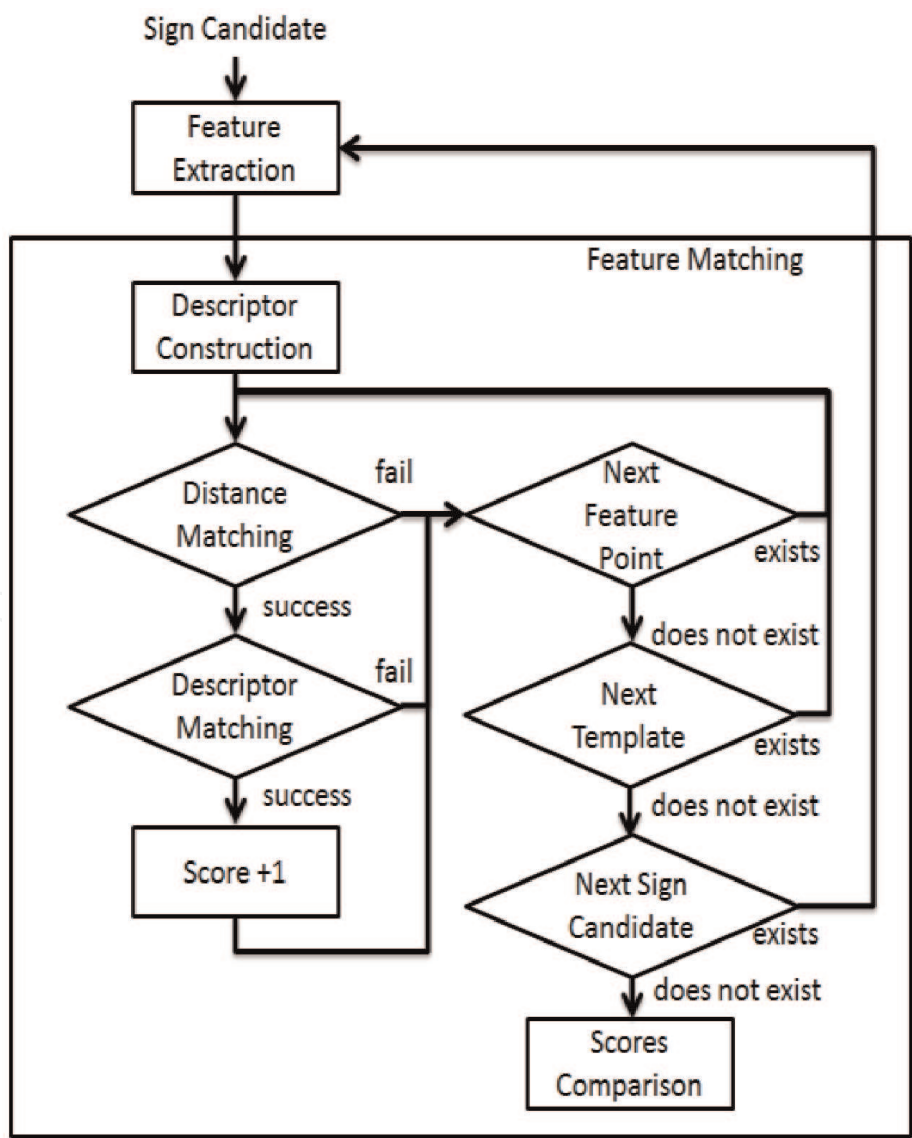nd Y representing the vectors of point x_i and y_i respectively, are randomly sampled with the Gaussian distribution and stored in a pre-built array to reduce the computation cost. They are sampled with Gaussian distribution stored in a pre-built array to reduce computation cost. To build a BRIEF descriptor, τ test is defined as in Eq. (27) and n is chosen as 256 to yield the best performance.

$$\tau(p; x, y) := \begin{cases} 1, p(x) < p(y) \\ 0, p(x) \geq p(y) \end{cases} \tag{27}$$

The advantages of BRIEF are obviously, low computation time and better matching performance whereas the disadvantage is that the BRIEF is not rotation invariant and scale invariant. Since the size of the detected signs is fixed and the road signs would not have too much rotation effect, these disadvantages do not influence the recognition result.

After descriptor construction, a two-step matching process comprising distance matching and descriptor matching is applied to match the detected sign candidates with the pre-built templates. Distance matching considers only the coordinates of the feature points. In this application of road signs recognition, the detected road signs should be a regular triangle sometimes with certain defects such as, lighting changes, slight rotation, and occluded with an object. Thus, the two similar feature points are not matched if the coordinates of these two feature points are different.

The goal of descriptor matching is to compute the distance between two descriptors, one is from the detected sign candidate and the other is from the pre-built template. As with all the binary descriptors, the measure of BRIEF distance is the number of different bits between two binary strings which can also be computed as the sum of the XOR operation between the strings.

After all sign candidates are matched, the scores comparison is applied to choose which template is the most suitable for final recognition result. The template with the scores higher than the others is judged as the result of the template matching. Moreover, the same result should be recognized a few times in several frames of a video to make sure that the result does not produce a false alarm.

The performance of the aforementioned DIP based algorithms in detecting and recognizing the road signs are discussed in the Section 3.

2.1.2 Computer neural network (CNN) methods to detect and recognize road signs

Artificial Neural networks (ANN) generally referred as Neural networks (NN), specifically as Computer Neural Networks (CNN) have been a sensation in the field of CV. ANNs, Artificial Intelligence (AI) and Deep Learning (DL) are interdependent and importantly, indispensable topics of recent years research and applications in engineering and in the technology industry. This reason for this prominence is that they currently provide the best solutions to many problems extensively in image recognition, speech recognition and natural language processing (NLP).

The inventor of one of the first neurocomputers, Dr. Robert Hecht-Nielsen defines a neural network as *"…a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."* In simpler words, the theme of ANNs are motivated by biological neural networks to learn and process the information fed to it. **Figure 28**
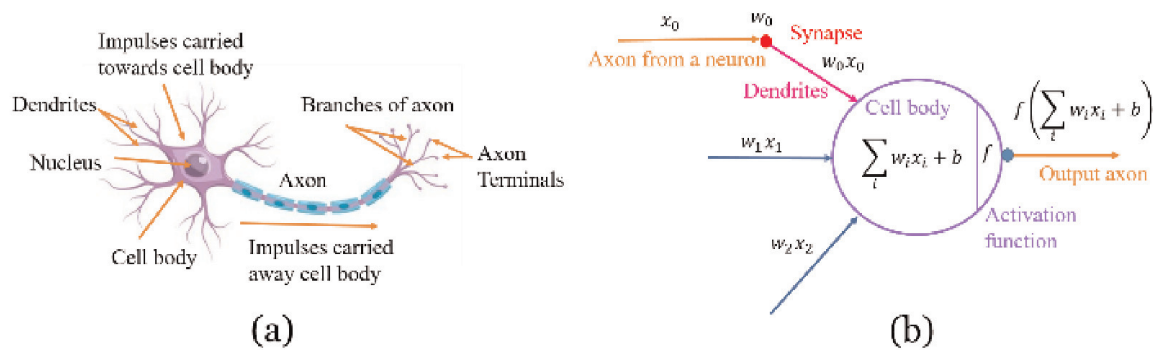
Figure 28.
(a) A representative biological neuron with (b) its mathematical model from [34].

shows the similarity in function of a biological neuron in **Figure 28(a)** with its respective mathematical model in **Figure 28(b)**.

A neuron is the fundamental unit of computation in a biological neural network whereas the basic unit of an ANN is called a node or a unit. The node/unit receives inputs from external sources and from other nodes within the ANN to process and computes an output. Every input has a characteristic weight (w) allotted based on its corresponding importance to other inputs. The node applies a function to the weighted sum of its inputs.

ANNs are generally organized in layers that are made up of numerous interconnected 'nodes' comprising an 'activation function' as in **Figure 29**. The inputs are presented to the ANN via the 'input layers', which communicates with one or more 'hidden layers' in which a particular processing is done by a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output. For the general model of ANN in **Figure 29**, the net input can be calculated as in Eq. (28) and the output by applying the activation function over the net input can be calculated using Eq. (29).

$$Y_{in} = X_1.w_1 + X_3.w_2 + ... X_n.w_n \; i.e., \text{the net input,} \; Y_{in} = \left(\sum i - n\right)X_i.w_i \quad (28)$$

$$Y = F(Y_{in}) \quad (29)$$

Recent researches and publications present that ANNs are extensively used for various applications ranging from object detections to learning to paint, to create images from the sentences, to play board games-AlphaGo and so on. There are many more unthinkable and unconvincing things done by the ANNs in the present days and research studies on further advancing them are going on rigorously.
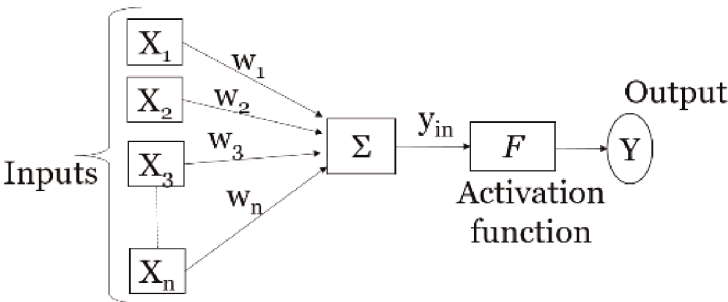

Figure 29.
The general model of an ANN.

23

There exist a various models presented from researches across the word for different applications. Some of the prominent CNNs are Single Shot Detector (SSD) [51], Region based Connect Neural Network (R-CNN) [52], Fast-RCNN [53], Faster-RCNN [54], You Only Look Once (YOLO) [55] and different versions of it, Generative Adversarial Networks (GAN) [56] and different modules [57] based on it and many more. This chapter also discusses CSPJacinto-SSD based on CSPNet [58] features in JacintoNet [59]. These innumerous ANNs are extensively used by the researchers and industries alike. The researchers and industries go hand-in-hand to investigate on further improvisations of the existing NNs, expanding them into diversified applications and solving a problem/challenge using effective and low-cost measures, ultimately manufacturing commercial products to make human lives easier and smarter.

In this chapter, we explore object detection NNs such as SSD, Faster-RCNN, YOLO and propose the newer CNN model termed as 'CSPJacinto-SSD' for the detection and recognition of road signs.

The SSD, as its name suggests, only need to take one single shot to detect multiple objects within the image. It has two components- a backbone and SSD head. The backbone model is a network used for pre-trained image classification. The SSD head is network with one or more convolutional layers added to the backbone network and the outputs are interpreted as the bounding boxes and classes of objects in the spatial location of the final layers activations as in **Figure 30**.

Faster-RCNN [54] comprises of two modules of which the first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector [53] that uses the proposed regions. The earlier version of the Faster-RCNN such as R-CNN and Fast R-CNN both use selective search method to find out the region proposals. The selective search method is a slow, hence time-consuming process affecting the performance of the network. To overcome these challenges, an advanced version of the R-CNN called Faster-RCNN was proposed [54] that has an object detection algorithm eliminating the selective search algorithm and the network learn the region proposals. **Figure 31** shows a Faster-RCNN network.

All of the previous object detection algorithms until the year 2015, used regions to locate the objects in the image. That means the network does not look at the complete image but only parts of the image that may the high probabilities of containing an object. In 2015, J. Redmon proposed a new NN called YOLO (You Only Look Once) [55] as in **Figure 32**. It is an object detection algorithm much different from the region-based algorithms. In YOLO, a single convolutional network predicts the bounding boxes and the class probabilities for these boxes.
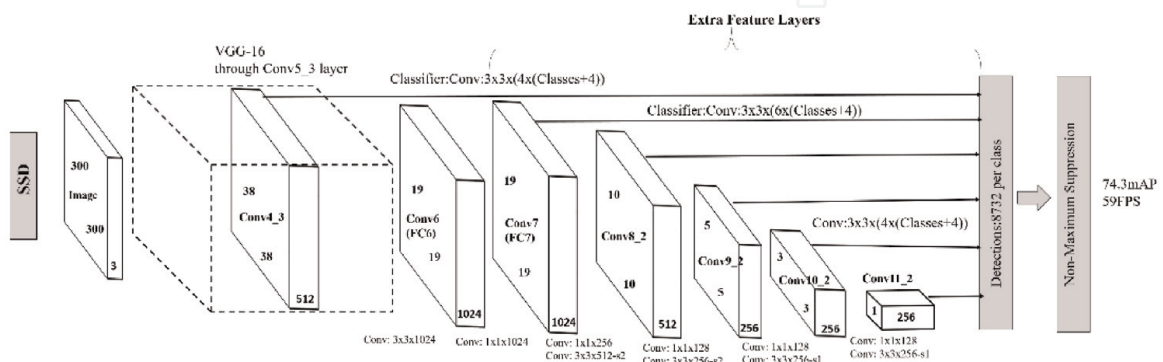

Figure 30.
SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences.
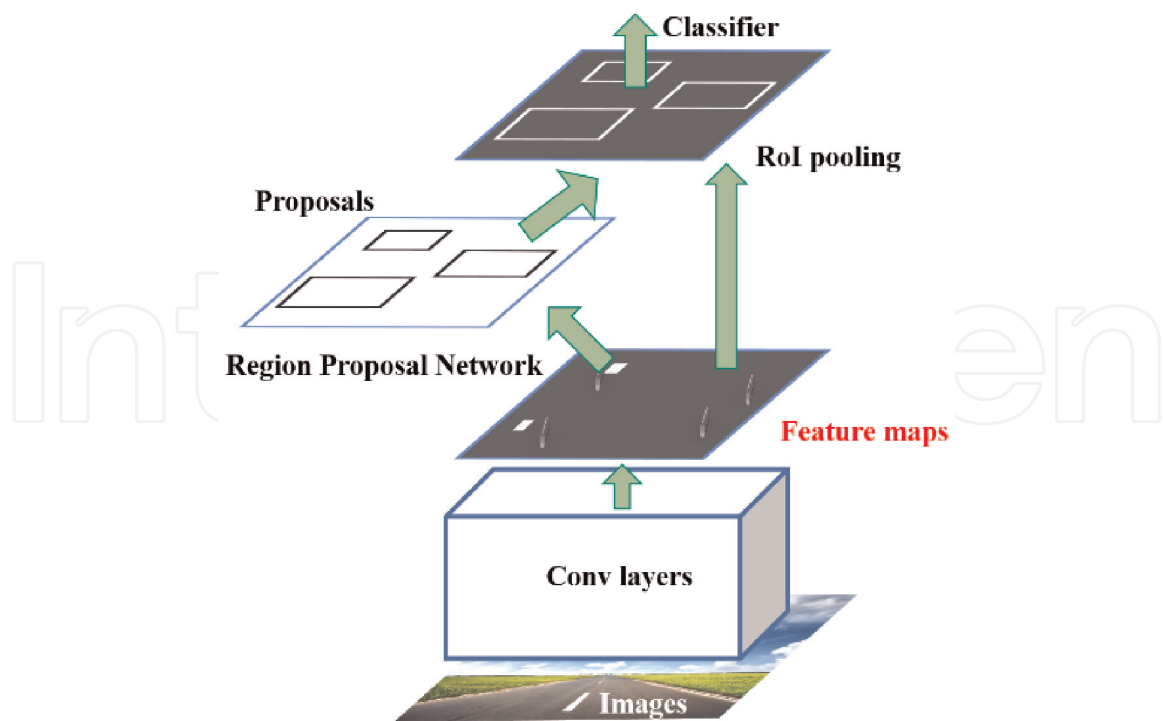
Figure 31.
A single, unified faster R-CNN for object detection.
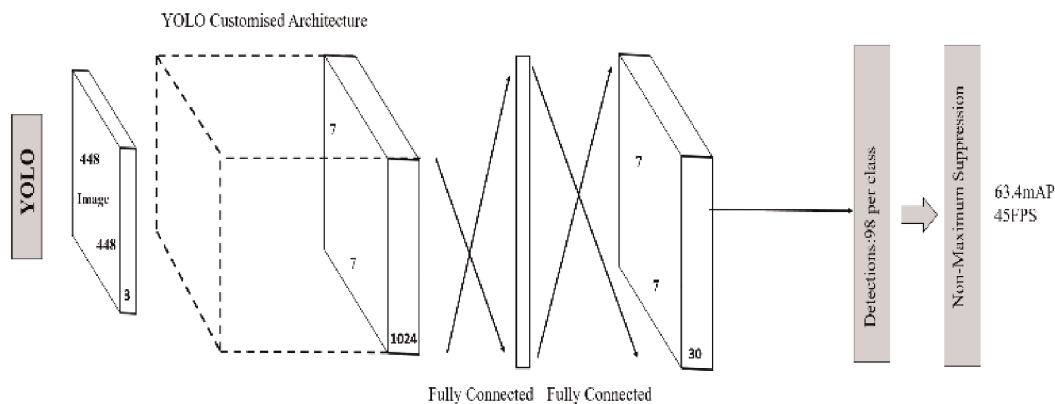
Figure 32.
A representative of the YOLO architecture for object detection.

The overall architecture of CSPJacinto-SSD is shown in the **Figure 33**. The CSPNet [58] features are added in JacintoNet [59], which is a simple light-weighted model composed of convolution, group convolution, and max-pooling layers. The Cross Stage Partial (CSP) feature is proved to improve the accuracy while reducing the model parameters and complexity. The function of CSP is simply to split the feature maps into two parts along channels at the input of each stage, one part sends into the convolution block, as usual, the other part skips all layers and concatenate with the output convolution block together as the final block output. In **Figure 33**, one blue and one green square can be seen as a convolution block. The blue arrows show the CSP feature as described above, and the red arrows show the output of each stage. The 1 x 1 convolution before the convolution block is used to increase the feature channels, and the 1 x 1 convolution after the convolution block is used to merge the context of features from the CSP layer. Out1 to Out5 labels the feature maps that are used for dense heads to process the bounding box outputs.

Figure 33.
CSPJacinto-SSD model architecture.

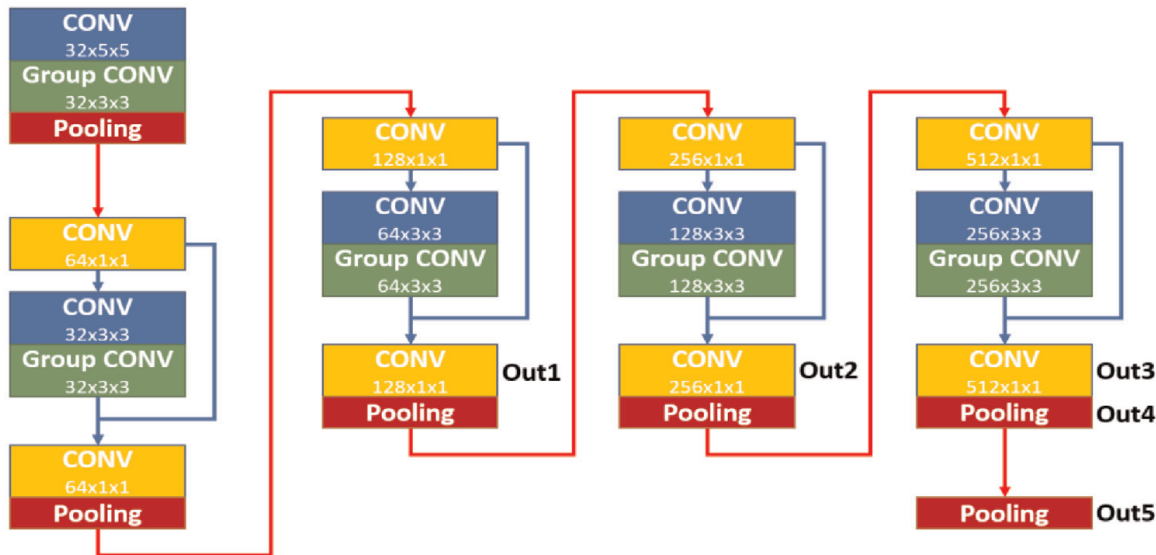|	Base sizes
Original SSD	16, 32, 64, 100, 300
Proposed CSPJacinto-SSD	16, 32, 64, 128, 256

Table 2.
Base size of anchor box.

The dense heads employed in the proposed CSPJacinto-SSD are referred to those in SSD, with some modifications in the anchor boxes based on the concept of multi-head SSD proposed in [60]. At dense head levels 2 to 4, there is an extra location of anchor boxes with offset 0, instead of only the original offset 0.5. This feature can increase the anchor boxes density improving the recall of object detection, especially used for light-weighted SSD models that need more anchor boxes to guide the objects' possibly appeared location.

The anchor box settings are a bit different from the original SSD model. The anchors 1:2 is changed to 1:1.5 because it may make the anchor borders denser, and preserve 1:3 anchors. The base size of the anchor boxes are modified compared to the original SSD, as shown in **Table 2**. Those anchor sizes can better fit with our model input size 256x256.

The performance of these ANN object detection networks in detecting and recognizing the road signs are discussed in the Section 3.

3. Results and discussion

3.1 The DIP based algorithms

The DIP based algorithms for detecting and recognizing road signs are implemented in C++ on a Visual Studio platform on desktop computer and a Freescale i.MX6. Due to the lack of standard video datasets dedicated to speed limit signs, we have examined the algorithm using the original video frames captured under different weather conditions such as daylight, backlight, cloudy, night, rain and snow.

3.1.1 System specifications

The DIP based algorithm discussed in Section 2.1 for speed limit and speed regulatory signs are realized on the standard desktop machine consisting of Intel® Core™ i7–3770 CPU, operating at 3.6GHz frequency with a memory of DDR3–1600-8GB on a Windows-7 64 bit and Ubuntu 14.04. The same DIP based algorithms are also realized on the Freescale i.MX 6 which is one of the standard developing processors for the real-time vehicular applications with a ARM Cortex-A9 CPU possessing an operating frequency of 1.2 GHz and memory of 1GB working with a Linux operation system. The same has Video Processing Unit (VPU) decoder-H.264, MPEG-4, H.263, MJPE and Image Processing Unit (IPU) possessing blending, rotating, scaling, cropping, de-interlacing, color spacing converting functions.

3.1.2 Performance: speed

On the desktop computer as well as on the Freescale i.MX 6, the size of images of speed limit signs is set to D1 resolution (720x480). The processing speed of the DIP based algorithm could reach up to 150fps on an average on the desktop computer and about 30fps on the Freescale i.MX 6. The image resolution is set as 1280x720 on the desktop computer and the performance can reach 161 fps on an average. On Freescale i.MX 6, the image resolution is also set as 1280x720 and the performance of the proposed algorithm is 17 fps.

3.1.3 Performance: accuracy and comparison

The performance accuracy of rectangular and circular speed limit signs and triangular speed regulatory signs' detection and recognition by DIP based algorithm discussed in Section 2.1 is tabulated in **Table 3**. The overall accuracy is defined as

	Rectangular speed limit road signs	Circular speed limit road signs	Triangular speed regulatory road signs
Video Resolution	720x480	720x480	1280x720
Total Video Frames	13187	14332	227445
Total Road Signs Count	77	113	60
Detected Signs	74	108	59
Detection Accuracy	**96.10%**	**95.58%**	**98.33%**
Total Detected Signs Frame Count	429	697	902
Total Detected Signs and Correctly Classified Frames	414	668	853
Total Correctly Recognized Signs Count	72	103	56
Recognition Accuracy	**97.30%**	**96.30%**	**94.92%**
Overall Accuracy (Detection Accuracy * Recognition Accuracy)	**93.51%**	**91.15%**	**93.33%**

Table 3.
The accuracies of the speed limit signs and speed regulatory signs detection and recognition.

"when a car passes a road scene with a road sign instance, the final output of the proposed algorithm is correct as that of the road sign visible to naked eyes."

The detection accuracy of rectangular speed-limit road signs is 96.10% and recognition accuracy is 97.30% accounting to the total accuracy of 93.51%. The detection accuracy of circular speed limit road signs is 95.58% and its recognition accuracy is 96.30% accounting to the overall accuracy of 91.15% whereas the detection accuracy of the triangular speed regulatory signs is 98.33% and the recognition accuracy is 94.92% resulting in the overall accuracy of 93.33%. The performance efficiency of these algorithms are evaluated under different weather conditions such as daytime, cloudy, with strong backlight, nighttime and during snow and rain. Some of these results are tabulated in **Tables 4–6**.

The efficiency of the proposed algorithm is also compared with the state-of-the-art works on the road signs detection and recognition.

As listed in **Table 7**, the proposed speed limit signs detection and recognition system is compared with some of the previous works. It can be implemented on embedded systems for real-time ADAS applications as it is capable of performing under computing resource and support both circular and rectangular speed limit road signs such as 15, 20, 25, 30, 35,....., 110 irrespective of the digit fonts from numerous countries adopting to the blob and breach features.

Video Sequence Number	1	2	3	4	5	6	7	8
Weather	Day	Day	Day	Day	Cloudy	Rain	Night	Night
Number of Signs	4	4	3	4	1	3	2	2
Detected signs	4	4	3	3	1	3	2	2
Missed Signs	0	0	0	1	0	0	0	0
Number of Frames with Sign Detection	15	17	10	9	4	12	7	9
Number of Correct Speed Limit Sign Recognition	15	16	10	8	3	12	6	9
Number of Wrong Speed Limit Sign Recognition	0	1	0	1	1	0	1	0

Table 4.
Some details of the rectangular speed limit road signs detection and recognition.

Video Sequence Number	1	2	3	4	5	6	7	8	9	10
Weather	Day	Day	Day	Day	Backlight	Cloudy	Snow	Rain	Night	Night
Number of Signs	4	6	5	5	5	4	3	2	2	2
Detected signs	4	6	5	5	5	4	2	2	1	2
Missed Signs	0	0	0	0	0	0	1	0	0	0
Number of Frames with Sign Detection	12	20	15	16	16	12	10	10	4	9
Number of Correct Speed Limit Sign Recognition	12	19	15	15	14	11	9	8	4	9
Number of Wrong Speed Limit Sign Recognition	0	1	0	1	2	1	1	2	0	0

Table 5.
Some details of the circular speed limit road signs detection and recognition.

Video Sequence Number	1	2	3	4	5	6	7
Weather	Day	Day	Day	Cloudy	Backlight	Night	Night
Number of Signs	6	3	5	2	4	3	4
Detected signs	5	3	5	2	3	2	4
Missed Signs	1	0	0	0	1	1	0
Number of Frames with Sign Detection	22	13	15	7	14	7	18
Number of Correct Speed Regulatory Sign Recognition	19	12	15	5	14	6	18
Number of Wrong Speed Limit Sign Recognition	3	1	0	2	0	1	0

Table 6.
Some details of the triangular speed regulatory road signs detection and recognition.

	[13]	[14]	[15]	[16]	[23]	[24]	**Speed Limit Road Signs**
CPU	2.16 GHz dual-core laptop	2.13 GHz dual-core laptop	2.13 GHz dual-core laptop	1.167GHz Intel Atom 230 and NVIDIA GeForce 9400 M GS GPU	2.26 GHz dual-core laptop	—	Intel Core i7–3770 3.40 GHz ARM Cortex-A9 1.2 GHz
Video Resolution	700 X 400	Image only	640 X 480	640 X 480	640 X 480	1920 X 1080	720 X 480
Frame Rate on PC	25fps	7.7fps (130 ms)	20fps	33fps	16fps	—	150fps
Detection Accuracy	87.00%	89.68%	97.50%	—	98.90%	—	95.84%
Recognition Accuracy	90.90%	88.97%	96.25%	95.00%	88.50%	96.24%	96.80%
Overall Accuracy	96.25%	90.90%	90.00%	88.00%	98.30%	94.00%	92.10%
Real-Time on Embedded System	X	X	X	X	X	X	O
Supports all types of Speed Limit Signs	X	X	X	X	X	X	O
Supports Different Digit Fonts on Speed Limit Signs	X	X	X	X	X	X	O

Table 7.
The comparison of the proposed speed regulatory road signs detection and recognition algorithm with previous works.

Table 8 lists the comparison of the proposed speed regulatory signs detection and recognition system with relative previous works. It can also be implemented on the embedded systems for real-time ADAS applications, as it is capable of performing under embedded computing resource and support different types of speed regulatory

signs as in **Figure 34** from numerous countries adopting feature extraction and feature matching features.

From the comparison listed in the **Tables 7** and **8**, it can be interpreted that DIP based algorithm discussed in this chapter is more robust i.e., it supports speed limit and speed-regulatory road signs of different types existing in most of the countries. It also performs well with the varied fonts of the speed limit signs and it can be compatibly implemented in an embedded system for the real-time applications. Importantly, it has decent accuracy when working with video as in the real camcorder environment compared to the state-of-the-art methods. Above all, the least complexity of our proposed algorithm yields higher fps compared to those of the other previous works.

Figure 35 shows the experimental results of the speed limit road signs detection and recognition method for detection and recognition of the rectangular speed limit road signs. **Figure 35(a-c)** is the result during the daytime, **Figure 35(d)** is during the cloudy weather, **Figure 35(e-f)** is during the rain and **Figure 35(g-j)** is during the nighttime.
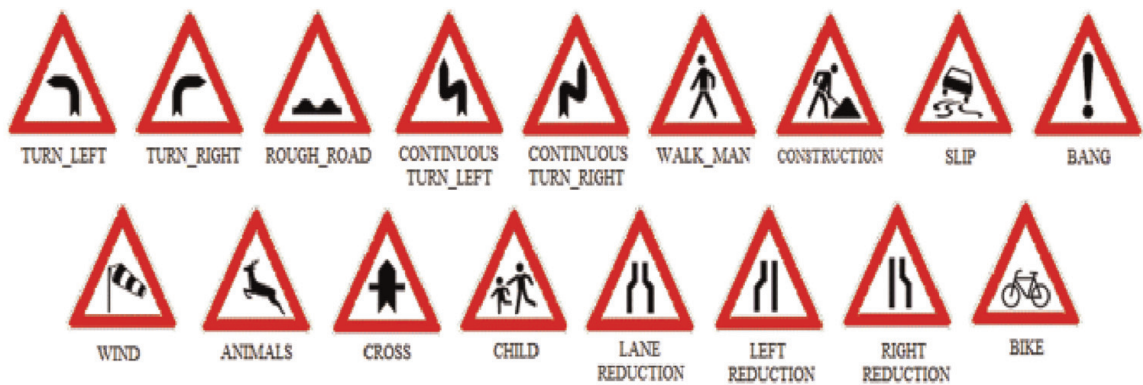

Figure 34.
Some samples of speed regulatory road signs.

	[17]	[18]	[19]	[20]	[21]	**Speed Regulatory Road Signs**
CPU	Intel Core i3	Pentium-IV 2.6 GHz	Intel Core i7	X	Tesla K20 GPU Platform	Intel Core i7–4790 3.60 GHz
Video Resolution	X	640 X 480	1292 X 964	640 X 480	1628 X 1236	1280 X 720
Frame Rate	0.5 fps	11.1 fps	33 fps	13.5 fps	27.9 fps	161 fps
Detection Accuracy	98.25%	97.7%	95.87%	96.00%	91.69%	98.33%
Recognition Accuracy	84.30%	93.60%	99.16%	96.08%	93.77%	94.49%
Overall Accuracy	82.82%	91.44%	95.07%	92.23%	85.97%	93.33%
Sensor Required	Vision	Vision	LIDAR + Vision	Vision	Vision	Vision

Table 8.
The comparison of the proposed speed regulatory road signs detection and recognition algorithm with previous works.

Figure 35.
The overall results for rectangular speed limit signs detection (a-c) during the daytime (d) during the cloudy weather (e-f) during the rain (g-j) during the nighttime.

Figure 36 shows the experimental results of the speed limit road signs detection method for detection as well as recognition of the circular speed limit road signs. **Figure 36(a-c)** is the result during the daytime, **Figure 36(d-e)** is during the backlight condition, **Figure 36(f-g)** is during the cloudy weather, **Figure 36(h-i)** is during the snow, **Figure 36(j-l)** is during the rains and **Figure 36(m-o)** is during the nighttime.

Figure 37 shows the experimental results of the speed regulatory road signs detection and recognition method for detection of the triangular speed regulatory road signs of which **Figure 37(a-c)** is the result during the daytime, **Figure 37(d-g)** is during the cloudy weather, **Figure 37(h-i)** is during the cloudy weather and **Figure 37 (j-l)** is during the nighttime.

Additionally, the proposed CV based method is capable of detecting and recognizing the speed limit signs ending with the digit "5" as in **Figure 38**.

3.2 The CNN based algorithms

The CNN based object detection algorithms such as SSD [51], Faster R-CNN [54], YOLO [55] and the proposed CSPJacinto-SSD are implemented in Python.
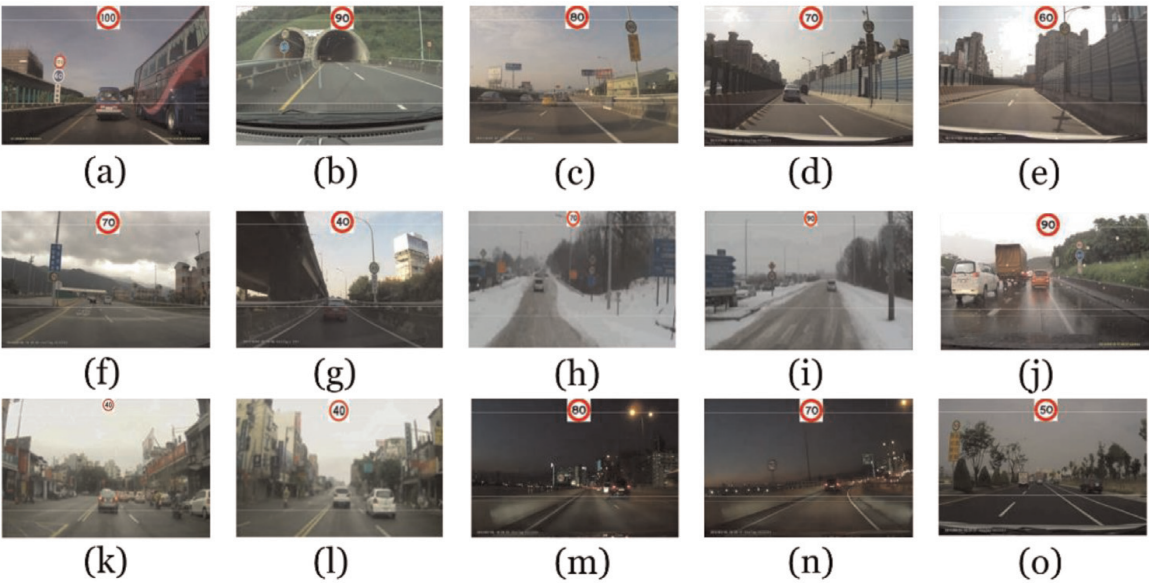

Figure 36.
The overall results for circular speed limit signs detection (a-c) during the daytime, (d-e) during the backlight condition, (f-g) during the cloudy weather, (h-i) during the snow (j-l) during rains, (m-o) during nighttime.
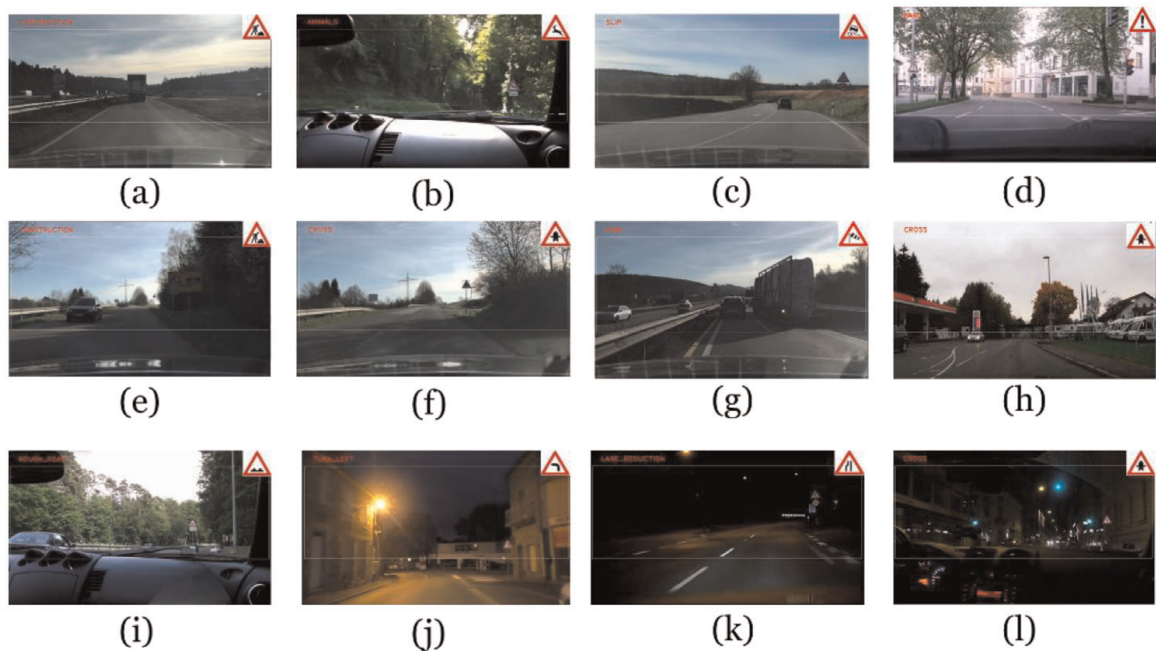
Figure 37.
The overall results for triangular speed regulatory road signs detection (a-c) during the daytime, (d-g) during the backlight condition, (h-i) during the cloudy weather (j-l) during the night.

Figure 38.
The detection and recognition result of speed limit ending with the digit '5'.

In order to carry out the roads signs detection and recognition, the same are trained and tested using a traffic signs dedicated dataset titled 'Tsinghua-Tencent 100 K' [9].

3.2.1 System specifications

The CNN based algorithm discussed in Section 2.1.2 for road signs are realized on the standard desktop machine consisting of Intel® Core™ i7–3770 CPU, operating at 4.2GHz frequency with a memory of DDR3–1600-16GB on a Windows-10 64 bit with Geforce GTX Titan X.

3.2.2 Performance: speed

On the desktop computer, the size of images are used as available in the dataset. The processing speed of the SSD, Faster-RCNN, YOLO and CSPJacinto-SSD object detection algorithms are around 20 fps, 5 fps, 21 fps, and 22 fps respectively.

3.2.3 Performance: accuracy and comparison

The performance efficiency of the CNN models is mostly calculated using mAP, AP and IoU as per [61]. Average precision (AP) is the most commonly used metric to measure the accuracy of object detection by various CNNs, and image-processing methods. The AP computes the average precision value for recall value. Precision measures how accurate the predictions are, by a method, i.e., the percentage of correct predictions whereas, Recall measures the extent to which the predicted positives are good. Eq. (30) is employed in this paper to estimate the AP where r refers to recall rate and $\hat{r}$ refers to the precision value for recall. The interpolated average precision [62] was used to evaluate both classification and detection. The intention in interpolating the precision/recall curve in this way is to reduce the impact of the wiggles in the precision/recall values, caused by small variations in the ranking. Similarly, the mean average precision (mAP) is the average of AP. The accuracy of these models [51, 54, 55] in detecting and recognizing road signs from dataset [9] is as tabulated in **Table 9**.

$$AP = \sum (r_{n+1} - r_n)\, p_{\text{interp}}\, (r_{n+1}) \text{ and } p_{\text{interp}}\, (r_{n+1}) = \max_{\hat{r} \geq rn+1} p(\hat{r}) \qquad (30)$$

4. The comparison of DIP and CNN based methods

The traditional DIP based methods are popular CV techniques namely SIFT, SURF, BRIEF, to list a few employed for object detection. Feature extraction process was carried out for image classification jobs. The features are descriptive of "interesting" in images. Various CV algorithms, such as edge detection, corner detection and/or threshold segmentation would be involved in this step. Thus extracted features from images forms the basis for definition for an object to be detected for respective class. During deployment of such algorithms, these definitions are sought in other images. If a significant number of features from defined for a class are found in other images, the image is classified, respectively.

Contrastingly, CNN came up with end-to-end learning where a machine learns about the object from a classes of annotated images which is termed as 'training' of a given dataset. During this, the CNN perceives the fundamental patterns in those

	Input resolution	mAP	FPS	Complexity per frame	No. of parameters
SSD 512	512 x 512	67.90%	~20	105.30G	42.40 M
Faster RCNN	608 x 608	75.20%	~5	120.60G	41.72 M
YOLO V4	608 x 608	71.40%	~21	64.40G	64.50 M
CSPJacinto-SSD	512 x 512	69.60%	~22	18.80G	11.40 M

Table 9.
Performance efficiency of CNN models in detection and recognition of road signs.

classes of images and consistently establishes a descriptive salient features for each specific classes of objects.

With almost all the researches and industries that involves CV are now employing CNN based methods, the functionalities of a CV professional has exceptionally changed in terms of both knowledge, skills and expertise as in **Figure 39**.

A comparison between the DIP and CNN based methods is tabulated in **Table 10**. The DIP based methods are more compatible and suited for the real-time applications of the ADAS as it is of low-complexity and does not require any data for pre-training of the system as compared to the data-hungry neural networks based systems. Apart from pre-training and complexity, the Freescale iMX6 consumes a total power of 1.17 W [64] in the video playback idle mode and other embedded systems would lie in the range where a minimum of 300 W [65] required by a basic GPU which makes them power-hungry as well. Additionally, the DIP based methods can perform object detection in any scenes irrespective of having seen the same or similar scene, but the CNN models can perform object detection of the objects that they models have seen
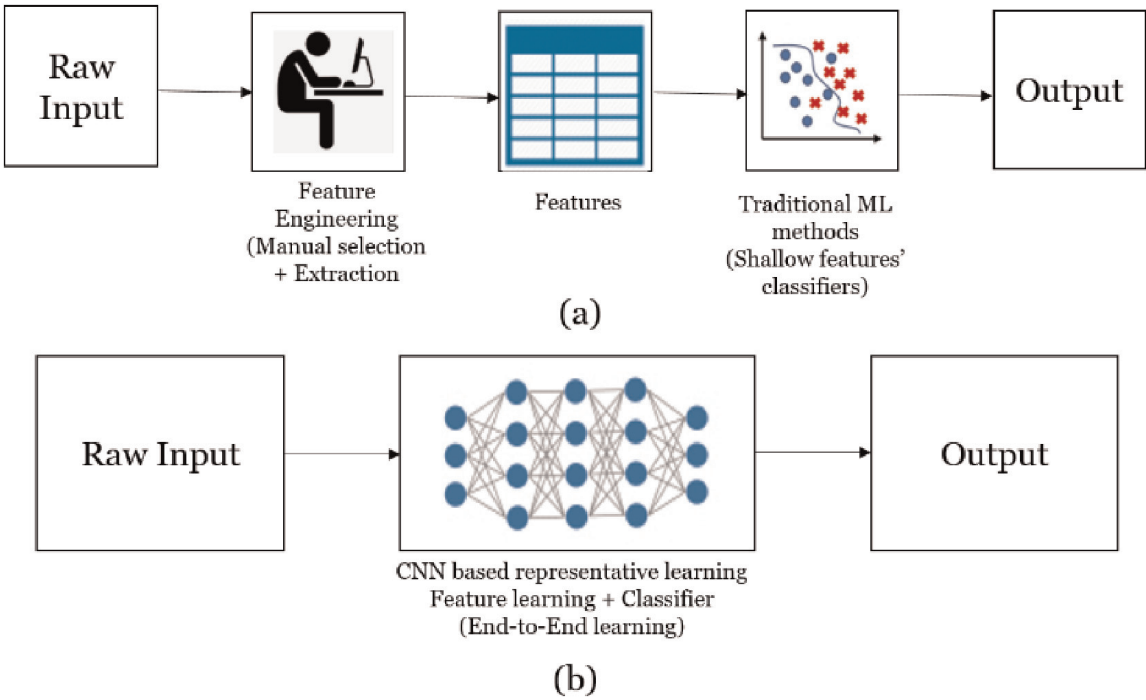

Figure 39.
Comparison of DIP and CNN based workflow. Fig. From [63].

Parameters	Proposed system	CNN based systems
Complexity	Low	High
Pre-training	X	O
Power Consumption	1.17 W	300 W
Recognition in scene never seen before	O	X
Robustness to inclement weathers	X	O

Table 10.
Comparison of the proposed system with that of CNN based systems.

34

during the training processes. Hence, the CNN models require good-amount of time spent to teach them perform a task followed by testing before employing them for the real-time applications unlike the DIP based methods. Moreover, CNN models exhibit high flexibility and perform better in inclement weathers than DIP based methods.

5. The conclusion

This chapter discussed traditional image processing methods and a few CNN based methods to detection and recognition of road signs for ADAS systems. It has been conclusive that DNNs perform better than the traditional algorithms with certain specific trade-offs with respect to computing requirements and training time. While there are pros and cons of both traditional DIP and CNN based methods, a lot of DIP based CV methods invented over the last 2–3 decades have now become obsolete because newer and much more efficinet methods of CNN have replaced them. However, knowledge and skills gained are also invaluable and not bounded by never inventions instead the knowledge of traditional methods forms a strong foundation for the professional to explore and widen his point-of-viewing a problems. Additionally, there are some of the traditional methods still being used in a hybrid-approach to improvise, innvoate leading to incredible innovations.

Acknowledgements

Conflict of interest

The authors declare no conflict of interest.

Author details

Vinay M. Shivanna[1]*, Kuan-Chou Chen[1], Bo-Xun Wu[1] and Jiun-In Guo[1,2,3]

1 Department of Electronics Engineering and Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu City, Taiwan

2 Pervasive Artificial Intelligence Research (PAIR) Labs, Hsinchu City, Taiwan

3 Wistron-NCTU Embedded Artificial Intelligence Research Center, Hsinchu City, Taiwan

*Address all correspondence to: vinay.ms23@gmail.com

IntechOpen

References

[1] J. Urry, "The 'System' of Automobility", Theory, Culture & Society, vol. 21, no. 4-5, pp. 25-39, October 2004.

[2] E. Eckermann, World History of the Automobile, Society of Automotive Engineers, Warrendale, PA, 2001.

[3] "Transportation: Motor Vehicle Accidents and Fatalities", The 2012 Statistical Abstract. U.S. Census Bureau, September. 2011.

[4] What is Machine Learning[Internet]? Ibm.com. Available from: https://www.ibm.com/cloud/learn/machine-learning

[5] What is Digital Image Processing (DIP) | IGI Global [Internet]. Igi-global.com. Available from: https://www.igi-global.com/dictionary/digital-image-processing-dip/48620

[6] Neural Network Definition[Internet]. Investopedia. Available from: https://www.investopedia.com/terms/n/neuralnetwork.asp

[7] How Artificial Intelligence Works [Internet]. Investopedia. Available from: https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp

[8] Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C. Detection of traffic signs in real-world images: The German traffic sign detection benchmark. The 2013 International Joint Conference on Neural Networks (IJCNN). IEEE; 2013. p. 1-8. Available from: http://10.1109/IJCNN.2013.6706807

[9] Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S. Traffic-Sign Detection and Classification in the Wild. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2016. p. 2110-2118. Available from: http://10.1109/CVPR.2016.232

[10] Deng J, Dong W, Socher R, Li L, Li K, Li F. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009. p. 248–255. Available from: http://10.1109/CVPR.2009.5206848.

[11] Everingham M, Van Gool L, Williams C, Winn J, Zisserman A. The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision. 2009;88(2):303-338.

[12] Mechanical Simulation [Internet]. Carsim.com. Available from: https://www.carsim.com/

[13] Jim Torresen, Jorgen W. Bakke and Lukas Sekanina, "Efficient Recognition of Speed Limit Signs," Proc. 2004 IEEE Intelligent Transportation Systems Conference, Washington, D.C., USA, October 3-6, 2004.

[14] Fabien Moutarde, Alexandre Bargeton, Anne Herbin, and Lowik Chanussot, "Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular Traffic Signs Recognition system," Proc. 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, June 13-15, 2007.

[15] Christoph Gustav Keller, Christoph Sprunk, Claus Bahlmann, Jan Giebel and Gregory Baratoff, "Real-time Recognition of U.S. Speed Signs," Proc. 2008 IEEE Intelligent Vehicles Symposium, June 4-6, 2008, The Netherlands.

[16] Wei Liu,Jin Lv, Haihua Gao, Bobo Duan, Huai Yuan and Hong Zhao, "An Efficient Real-time Speed Limit Signs Recognition Based on Rotation Invariant Feature", Proc. 2011 IEEE Intelligent Vehicles Symposium (IV) Baden-Baden, Germany, June 5-9, 2011.

[17] Zumra Malik and Imran Siddiqi, "Detection and Recognition of Traffic Sign Road Scene Images", 12th International Conference on Frontiers of Information Technology, pp 330-335, Dec 17-19, 2014.

[18] Vavilin Andrey and Kang Hyun Jo, "Automatic Detection and Recognition of Traffic Signs using Geometric Structure Analysis", International Joint Conference on SICE-ICASE, Oct 18-21, 2006.

[19] Lipu Zhou, Zhidong Deng, "LIDAR and Vision-Based Real-Time Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicle", IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), Oct 8-11, 2014.G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[20] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel, "Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark" The International Joint Conference on Neural Networks (IJCNN), 2013.

[21] M. liang, M. Yuan, X. Hu, J. Li, and H. Liu, "Traffic sign detection by supervised learning of color and shape," in Proceedings of IEEE International Joint Conference on Neural Networks, 2013.

[22] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool, "Traffic sign recognition - how far are we from the solution?" in Proceedings of IEEE International Joint Conference on Neural Networks, 2013.

[23] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A robust, coarse-to-fine traffic sign detection method," in Proceedings of IEEE International Joint Conference on Neural Networks, 2013.

[24] Supreeth H.S.G, Chandrashekar M Patil, "An Approach Towards Efficient Detection and Recognition of Traffic Signs in Videos using Neural Networks" International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp 456-459

[25] Nadra Ben Romdhane, Hazar Mliki, Mohamed Hammami, "An mproved Traffic Signs Recognition and Tracking Method for Driver Assistance System", IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)-2016

[26] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", The International Joint Conference on Artificial Intelligent, 1981, pp. 674–679.

[27] Y. Zhang, C. Hong and W. Charles, "An efficient real time rectangle speed limit sign recognition system," 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, 2010, pp. 34-38. DOI: 10.1109/IVS.2010.5548140

[28] A. Mammeri, A. Boukerche, J. Feng and R. Wang, "North-American speed limit sign detection and recognition for smart cars," 38th Annual IEEE Conference on Local Computer Networks - Workshops, Sydney, NSW, 2013, pp. 154-161

[29] C. Wang, "Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning," 2018 International Conference on Robots & Intelligent System (ICRIS), 2018, pp. 150-152, doi: 10.1109/ICRIS.2018.00047.

[30] R. Hasegawa, Y. Iwamoto and Y. Chen, "Robust Detection and Recognition of Japanese Traffic Sign in the Complex Scenes Based on Deep Learning," 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), 2019, pp. 575-578, doi: 10.1109/GCCE46687.2019.9015419.

[31] Y. Sun, P. Ge and D. Liu, "Traffic Sign Detection and Recognition Based on Convolutional Neural Network," 2019 Chinese Automation Congress (CAC), 2019, pp. 2851-2854, doi: 10.1109/CAC48633.2019.8997240.

[32] Y. Yang, H. Luo, H. Xu and F. Wu, "Towards Real-Time Traffic Sign Detection and Classification," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 7, pp. 2022-2031, July 2016, doi: 10.1109/TITS.2015.2482461.

[33] R. Jain and D. Gianchandani, "A Hybrid Approach for Detection and Recognition of Traffic Text Sign using MSER and OCR," 2018 2nd International Conference on I-SMAC, 2018, pp. 775-778, doi: 10.1109/I-SMAC.2018.8653761.

[34] M. Z. Abedin, P. Dhar and K. Deb, "Traffic sign recognition using hybrid features descriptor and artificial neural network classifier," 2016 19th International Conference on Computer and Information Technology (ICCIT), 2016, pp. 457-462, doi: 10.1109/ICCITECHN.2016.7860241.

[35] Lin Y, Chou T, Vinay M, Guo J. Algorithm derivation and its embedded system realization of speed limit detection for multiple countries. 2016 IEEE International Symposium on Circuits and Systems (ISCAS). Montreal, QC: IEEE; 2016. p. 2555-2558. Available from: http://10.1109/ISCAS.2016.7539114

[36] Chou T, Chang S, Vinay M, Guo J. Triangular Road Signs Detection and Recognition Algorithm and its Embedded System Implementation. The 21st Int'l Conference on Image Processing, Computer Vision and Pattern Recognition. CSREA Press; 2017. p. 71-76. Available from: http://ISBN: 1-60132-464-2

[37] Gareth Loy and Nick Bames, "Fast Shape-based Road Sign Detection for a Driver Assistance System," Proc. IEEE/RSl International Conference on Intelligent Robots and Systems, September 28 - October 2, 2004.

[38] Nick Barnes and Gareth Loy, "Real-time regular polygonal sign detection", Springer Tracts in Advanced Robotics Volume 25, pp 55-66, 2006.

[39] Sebastian Houben, "A single target voting scheme for traffic sign detection," Proc. 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, June 5-9, 2011.

[40] Feature Detectors - Sobel Edge Detector. Homepages.inf.ed.ac.uk. Available from: https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm

[41] Fatin Zaklouta and Bogdan Stanciulescu, "Real-time traffic sign recognition in three stages," Robotics and Autonomous Systems, Volume 62, Issue 1, January 2014.

[42] Derek Bradley and Gerhard Roth, "Adaptive Thresholding using the Integral Image," Journal of Graphics,

GPU, and Game Tools, Volume 12, Issue 2, 2007.

[43] Alexandre Bargeton, Fabien Moutarde, Fawzi Nashashibi, and Benazouz Bradai, "Improving pan-European speed-limit signs recognition with a new "global number segmentation" before digit recognition," Proc. 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, June 4-6, 2008.

[44] Dang Khanh Hoa, Le Dung, and Nguyen Tien Dzung, "Efficient determination of disparity map from stereo images with modified Sum of Absolute Differences (SAD) algorithm", 2013 International Conference on Advanced Technologies for Communications (ATC 2013)

[45] J. R. Parker, "Vector Templates and Handprinted Digit Recognition," Proc. 12th IAPR International Conference on Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision & Image Processing, Jerusalem, 9-13 Oct 1994.

[46] Phalgun Pandya and Mandeep Singh, "Morphology Based Approach To Recognize Number Plates in India," International Journal of Soft Computing and Engineering (IJSCE), Volume-1, Issue-3, July 2011.

[47] Kamaljit Kaur and Balpreet Kaur, "Character Recognition of High Security Number Plates Using Morphological Operator," International Journal of Computer Science & Engineering Technology (IJCSET), 2011 IEEE Intelligent Vehicles Symposium (IV), Vol. 4, May, 2013.

[48] Lifeng He and Yuyan Chao, "A Very Fast Algorithm for Simultaneously Performing Connected-Component Labeling and Euler Number Computing", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 24, NO. 9, SEPTEMBER 2015

[49] Rachid Belaroussi and Jean Philippe Tarel, "Angle Vertex and Bisector Geometric Model for Triangular Road Sign Detection", IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1-7, 2009.

[50] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features", European Conference on Computer Vision, May 2006.

[51] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C et al. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV). 2016. p. 21–37. Available from: http://arXiv: 1512.02325

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

[53] Girshick R. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV) [Internet]. IEEE; 2015. Available from: http://10.1109/ ICCV.2015.169

[54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, 2014, pp. 2672– 2680.

[55] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017; 39(6):1137-1149.

[56] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair,

A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, 2014, pp. 2672– 2680.

[57] Martin Arjovsky, Soumith Chintala, and LeonBottou.Wasserstein GAN. arXiv:1701.07875v2 [stat.ML], 9 Mar 2017.

[58] C. Wang, H. M. Liao, Y. Wu, *P. Chen*, J. Hsieh, and I. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 14-19 June 2020 2020, pp. 1571-1580.

[59] M. Mathew, K. Desappan, P. K. Swami, and S. Nagori, "Sparse, Quantized, Full Frame CNN for Low Power Embedded Devices," in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 21-26 July 2017 2017, pp. 328-336

[60] C. Y. Lai, B. X. Wu, T. H. Lee, V. M. Shivanna, and J. I. Guo, "A Light Weight Multi-Head SSD Model For ADAS Applications," in 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), 3-5 Dec. 2020 2020, pp. 1-6

[61] Jonathan, H. mAP (mean Average Precision) for Object Detection. Available online: **https://medium.com/ @jonathan_hui/map-mean-average- precision-for-object-detection-45c 121a31173** (accessed on 12 July 2018).

[62] Salton, G.; McGill, M.J. *Introduction to Modern Information Retrieval*; McGraw-Hill: New York, NY, USA, 1986. [**Google Scholar**]

[63] Wang J, Ma Y, Zhang L, Gao RX (2018) Deep learning for smart

manufacturing: Methods and applications. J Manuf Syst 48:144–156. https://doi.org/10.1016/J.JMSY. 2018.01.003

[64] Freescale Semiconductor, "i.MX 6Dual/6Quad Power Consumption Measurement" from "https://bit.ly/ 2ATVcWk"

[65] GEFORCE. "Desktop GPUs- Specifications" from "https://www. geforce.co.uk/hardware/desktop- gpus/geforce-gt-1030/specifications"