Detection and recognition of U.S. speed signs from grayscale images for intelligent vehicles

Pradeep Kumar Kanaparthi

The University of Toledo

Follow this and additional works at: http://utdr.utoledo.edu/theses-dissertations

Recommended Citation
http://utdr.utoledo.edu/theses-dissertations/345

This Thesis is brought to you for free and open access by The University of Toledo Digital Repository. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of The University of Toledo Digital Repository. For more information, please see the repository’s About page.
A Thesis

entitled

Detection and Recognition of U.S. Speed Signs from Grayscale Images for Intelligent Vehicles

by

Pradeep Kumar Kanaparthi

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Electrical Engineering

_________________________________________
Dr. Ezzatollah Salari, Committee Chair

_________________________________________
Dr. Junghwan Kim, Committee Member

_________________________________________
Dr. Jackson Carvalho, Committee Member

_________________________________________
Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo

December 2012
The aim of this thesis is to develop and implement an algorithm that automatically detects and recognizes U.S. speed signs, from the grayscale images captured by a camera mounted on the interior mirror of a vehicle, as a part of designing smarter vehicles. The system operates in real-time within the computational limits of contemporary embedded general purpose processors. This system will assist the driver by providing the necessary information, regarding the assigned speed limits, right in front of him and provide additional safety measures by monitoring the vehicle’s speed.

The proposed method consists of two phases in it: a detection phase, in which all the possible speed signs in the input image are detected first, and a recognition phase, in which the detected regions are recognized and the information regarding the speed limits is extracted from them. The detection phase utilizes the region characteristics, such as aspect ratio and size, to hypothesize the speed sign locations in the input image. We have utilized the idea of connected component labeling technique and adapted it for the grayscale images, to divide the input image into a set of regions. The recognition phase calculates the invariant features of the inner parts of the detected regions using Hu’s
moments. It verifies the hypothesis first, before extracting the assigned speed limit from the detected region using a feed forward neural network. The proposed method was experimented on a number of traffic images and the results show that the region characteristics are more immune to different noisy conditions such as partial occlusions, cluttered backgrounds and deformations.

**Keywords:** Speed sign, connected component labeling, regions, optical character recognition, neural network.
To my father and my lovely mother. Without my family members’ love, support, belief and hope I would not reach this point in my life.
Acknowledgements

First, I would like to thank god for the family I have, people I met, and the knowledge I gained.

I am very grateful to Dr. Ezzatollah Salari for letting me working under his supervision. Dr. Salari supervision and guidance boost the qualities of my engineering and research skills. Without his guidance and persistent help, this dissertation would not have been possible. I am lucky to be one of his students.

Special thanks to the distinguished faculty members who served on my committee: Dr. Junghwan Kim and Dr. Jackson Carvalho.
# Table of Contents

Abstract ................................................................................................................................. iii
Acknowledgments .................................................................................................................. vi
Table of Contents ................................................................................................................ v
List of Tables ......................................................................................................................... ix
List of Figures ....................................................................................................................... x

1. Introduction ......................................................................................................................... 1
   1.1. Background ................................................................................................................... 1
   1.2. Necessity of Automatic Speed Detection System ....................................................... 2
   1.3. Algorithm and Contributions ...................................................................................... 3
   1.4. Outline of the Thesis ................................................................................................. 5

2. Literature Review .............................................................................................................. 6
   2.1. Traffic Sign Detection Algorithms .............................................................................. 6
   2.2. Traffic Sign Recognition Algorithms ........................................................................ 8

3. Hypothesizing the Speed Sign Regions in the Input Traffic Image .............................. 10
   3.1. Introduction to Connected Component Labeling ...................................................... 11
   3.2. Original Two-pass Connected Component Labeling Algorithm ........................... 12
   3.3. Modified Connected Component Labeling Algorithm for Grayscale Images ...... 14
3.3.1. Resolving Label Equivalence using Union-find Algorithm .................. 15

3.4. Hypothesizing the Speed Sign Regions Using Region Parameters .............. 16

4. Recognizing the Hypothesized Regions ............................................................. 18

4.1. Introduction to Object Recognition ................................................................. 18

4.2. Steps Involved in the Proposed Recognition Algorithm ................................. 20

4.2.1. Pre-processing ............................................................................................... 21

4.2.2. Feature Extraction from Hu's Moments ......................................................... 22

4.2.3. Region Classification and Speed Information Extraction ......................... 25

5. Test Results and Analysis ...................................................................................... 31

5.1. Experiment 1 ....................................................................................................... 32

5.1.1. Detection Phase ............................................................................................ 32

5.1.2. Recognition Phase ......................................................................................... 35

5.2. Experiment 2 ....................................................................................................... 38

5.3. Experiment 3 ....................................................................................................... 41

5.4. Experiment 4 ....................................................................................................... 45

5.5. Experiment 5 ....................................................................................................... 48

6. Conclusions and Future Work ............................................................................. 53

References .................................................................................................................. 55
List of Tables

5.1 Examples of few regions’ characteristics from the figure 5-2 ................................................. 34
5.2 Classifying the region by recognizing its components .......................................................... 36
5.3 Some of the regions parameters from the labeled image ....................................................... 39
5.4 Classifying the hypothesized regions in the figure 5-12 ....................................................... 40
5.5 Region parameters from the labeled image .............................................................................. 43
5.6 Classifying the hypothesized region in the figure 5-17 ....................................................... 44
5.7 Region parameters from the image 5-21 .................................................................................. 46
5.8 Classifying the hypothesized region in the figure 5-22 ....................................................... 47
5.9 Region parameters from the labeled image in figure 5-26 ................................................... 50
5.10 Classifying the hypothesized regions in the figure 5-27 ..................................................... 51
List of Figures

1-1 Block diagram of automatic speed detection system ........................................ 3
1-2 Flow chart of the algorithm ............................................................................... 4
3-1 Sample traffic scene ............................................................................................ 10
3-2 Forward raster scan mask ..................................................................................... 13
3-3 Label equivalent matrix ....................................................................................... 15
3-4 Resolved label equivalent matrix ......................................................................... 16
4-1 (a) Normalized image (b) Inverted black and white image (c) Labeled image ...... 21
4-2 Architecture of the multilayer neural network ..................................................... 27
5-1 Original image ....................................................................................................... 32
5-2 Grayscale image labeling ..................................................................................... 33
5-3 Histogram of the labeled image showing the sizes of the regions ...................... 34
5-4 Generated hypothesis ............................................................................................ 35
5-5 (a) Detected region (b) Inverted black and white image (c) Labeled image ........ 35
5-6 Analyzing the upper half of the detected region .................................................. 36
5-7 Analyzing the lower half of the region .................................................................. 37
5-8 Recognized speed limit ......................................................................................... 37
5-9 Original image ....................................................................................................... 38
5-10 Labeled image .................................................................................................... 38
5-11 Histogram of the labeled image showing the sizes of the regions ..................... 39
Chapter 1

Introduction

1.1. Background

Our universe is filled with different kinds of rules or laws such as physical laws, environmental laws etc. Similarly we have created a set of traffic rules, to guide the people travelling and to regulate the traffic flow. Some of these traffic rules are indicated using the sign boards positioned by the sides of the roads. There are different kinds of sign boards used in America, such as yield sign, stop sign, speed-limit sign, warning sign and guide sign. These sign boards are designed in such a way that by looking at them the driver could extract the information from it. The driver requires all kinds of information while travelling. A driver does multiple tasks while he is driving such as, monitoring the road path ahead, observing the vehicles around him, constantly checking the blind spots, indicating the lane changes and keeping in track with the traffic rules. Thus the driver is overburdened while driving. Nowadays the new cars are being equipped with different kinds of new technologies which help the driver in his driving duties. The topic that comprises all such kinds of technologies is known as driver assistance system. As the name suggests it assists the driver by providing the required information for him, warning about the traffic violations and avoiding the accidents. It includes subtopics like sign
board detection, monitoring a lane change, vehicle detection, auto parking and speed
detection. A sign board detection system detects the sign boards that are present along the
sides of the roads, recognizes them and displays them on the vehicle’s dash board. A lane
change monitoring system assists the driver by warning him if the blind spots are not
clear while he is changing the lanes. A vehicle detection system automatically detects the
vehicles in the road path ahead. An auto parking system parks the vehicle for the driver.
A speed detection system detects the speed-limit sign board located by the side of the
road, recognizes it and displays the speed limit on the dash board in front of the driver.
Thus these tools can help the driver by reducing his efforts while driving and provide
additional safety measures.

1.2. Necessity of Automatic Speed Detection System

In United States speed limits vary depending on the locations and road conditions.
The speed limits on the freeways and other city roads are different. The work zones have
different speed limits. In some locations speed limits are different for day timings and
night timings. So in order to follow the traffic rules the driver has to look for the speed
limit signs located by the sides of the roads. If the vehicle is equipped with the automatic
speed detection system it would detect the speed-limit signs, recognizes them and
displays the speed limit information right in front of the driver. This system would also
be helpful to warn the driver if he is driving much faster than the assigned speed limit. By
interconnecting the speed detection system and the cruise control system, the cruising
speed could be automatically chosen.
Nowadays the new vehicles are being equipped with the navigation systems which also provide the speed limit information. But, according to the automaker BMW, this information is only 70 percent accurate. The work zone speed limits and other temporary speed limits are not provided by these navigation systems. BMW also claims that by using the speed detection system along with the navigation system the accuracy of the information could be increased to 95 percent.

1.3. Algorithm and Contributions

Figure 1-1 shows the global view of the automatic speed detection system. A camera mounted on the rear view mirror of the vehicle is used to capture the traffic scenes that appear in front of the vehicle. Each traffic scene is processed in real-time using the proposed algorithm to detect the speed-limit sign board present in it and to extract the speed-limit information from it. The extracted speed-limit information is displayed on the dash board of the vehicle.

![Block diagram of automatic speed detection system](image)

Figure 1-1 Block diagram of automatic speed detection system

The proposed method for the speed detection system consists of two major parts, hypothesizing the sign board locations in the input image and recognizing the hypothesized regions and extracting the useful information from them. Each part has multiple steps involved in it. The flow chart of the proposed algorithm is shown in the figure 1-2. The proposed method operates on grayscale images, detects the speed signs
first and then recognizes them to verify the hypothesis and to extract the speed limits from the detected regions. These steps are discussed in detail in the following chapters.

![Flow chart of the algorithm](image)

**Figure 1-2 Flow chart of the algorithm**

Our contributions include addressing the problems faced for the automatic speed limit detection system. The following are the problems faced for the speed detection system. The shape of the sign board changes as the viewing angle changes. There is a lot of background noise present in the traffic scenes making the detection process very difficult. Some of the sign boards are deformed, rotated and tilted. Apart from these there are situations like partial occlusions of the sign boards and different lighting conditions. Our contributions include the efforts to handle these situations,
• The original connected component labeling algorithm is modified to apply it on the grayscale images.
• The proposed algorithm is able to detect the speed-limit signs under deformations, different perceptions and rotation.
• More robust features of the speed-limit sign board are utilized in detecting them.
• Detection of the sign board under partial occlusions is also achieved.
• We have empowered the neural networks with a rejection mechanism to reject the spurious patterns in the recognition phase.
• We have utilized the spatial features of the speed-limit sign board to reduce the computations required in the recognition phase of the algorithm.

In brief, the proposed algorithm detects the speed-limit signs automatically and recognizes them. The most common problems faced in the process like occlusions, different perceptions, tilted sign boards and different lighting conditions are solved using the proposed algorithm.

1.4. Outline of the Thesis

The remaining part of the thesis is organized as follows. Chapter 2 discusses the literature work. Chapter 3 explains the detection phase of the algorithm. Chapter 4 explains feature extraction method, classification method and speed limit extraction using artificial neural networks. Chapter 5 produces the results obtained in each phase. And finally conclusions and future work is noted in chapter 6.
Chapter 2

Literature Review

The intelligent driver assistance systems are an interesting area of research, because it comprises of topics which are real and very common. For example, the auto parking system has been successfully implemented in new vehicles. Similarly the other topics like sign board detection, lane change detection and speed detection can be dealt using existing technologies to provide extra comfort and safety to the driver. A lot of research work is going on in these topics. The speed limit detection system employs two most important image processing methods namely object detection and object recognition. The existing algorithms in these areas, used for our application, are discussed separately in the following sections.

2.1. Traffic Sign Detection Algorithms

The speed limit detection algorithm first detects the speed limit sign board present in the input image. Various object detection algorithms exist in the literature. Different authors have implemented different algorithms utilizing different features of the objects. Some of the key features of the objects used are shape, color, symmetry, aspect ratio, size and edges. We will discuss the algorithms that used these features for object detection.
Shape is an efficient tool to describe an object and is used by many authors to locate a traffic sign board in the input image. Fabien and Alexandre [1] used a circular Hough-transform for detecting European speed-limit signs and a rectangle detection method for detecting American speed-limit signs. Angela Tam and Hua Shen [2] also used Hough-transform for the detection of quadrilateral sign boards. Xiaoou Tang and Hua Shen [3] used a gradient based Hough transform to detect the edges in the image and then a close circuit detection and redundant line deletion technique to locate the sign board in the image. Loy and Barnes [4] presented an algorithm that detects polygon-shaped signs (e.g., square, triangle, octagon) using radial symmetry detector. The accuracy of these shape based methods depends a lot on the edge operators being used by them and these edges are vulnerable to noisy pixels.

Some papers detected sign boards by using their color information. These papers assumed that the color information captured by the camera is invariant under varying lighting conditions. Many algorithms exist in this category (e.g., [5], [6], [7], [8]). These authors have utilized different color spaces to strengthen their assumption of color invariance.

The color and shape features of the sign boards are both utilized in some algorithms. N. Kehtaranavaz, N.C. Griswold and D.S. Kang [9] initially converted the RGB image into its HSI color space, segmented the resulting image and then performed shape analysis to detect the sign boards in the images.

All the sign boards are designed to have bilateral symmetrical shapes. Many authors have designed different algorithms making use of the symmetrical shape features of the sign boards. Zelinsky and Barnes [10] used the symmetry feature of the signs to
detect circular signs with a measure of radial symmetry in the image. In another paper Nick Barnes [11] used radial symmetry and regular polygon detection algorithm after reducing the noise in the images to improve the detection results. But it is observed that these symmetry features of the sign board, when measured from the images, are inconsistent under different object transformations and different perceptions.

Sign board detection is also performed using template matching methods. But because they are computationally expensive they are applied on smaller areas determined by performing color segmentation before using template matching [8]. A fast template matching method using simulated annealing algorithm is proposed by Betke and Makris [12].

Some methods make use of the prior knowledge about the image formation. For example if we assume a straight road path in the traffic scenes, it would eliminate large portions of the non-sign board regions in the images. But these methods will not succeed in the curved paths. Some methods are developed based on corner detection techniques (e.g.,[13],[14],[15],[16]). Neural network based detection algorithms like [17] also exist in the literature.

2.2. Traffic Sign Recognition Algorithms

The detected regions in the first phase are recognized in the recognition phase. Object recognition algorithms generally involve two important steps; feature extraction and classification. Feature extraction methods focus on calculating the important features of the objects that are invariant under different conditions. And classification methods aim at distinguishing different feature patterns and classifying the objects accurately.
Features are calculated for the structural parts of the detected regions. Many algorithms exist in the literature, which calculated invariant properties of the printed characters. For example the geometric moment invariants derived by Hu [18] are used in many optical character recognition algorithms. Zernike [19] proposed a set of moments based on Zernike polynomials. Juan J. Rodriguez and Jesus Maudes [20] used the number of holes in the characters and their horizontal and vertical projections as features. Hsien-Chu Wu, Chwei-Shyong TSAI and Ching-Hao LAI [21] proposed character recognition algorithm based on pattern mapping. They derived the features from the histograms of the different parts of the characters.

Hausdorff [22] distance, which is used to compare two binary images, is also employed in character recognition algorithms. We also have algorithms based on support vector machines [23] and template matching (e.g., [24], [25]) techniques. Authors of other papers, who employed Hidden Markov models (e.g., [26],[27]), claim that they are able to achieve recognition results of 95.7%.

Artificial neural network (e.g., [28], [29], [30], [31]) is another popular method used for optical character recognition in many applications. In [32] probabilistic neural networks method is employed for recognizing the alphabets and numbers. J.R. Parker and Pavol Federl [33] used genetic algorithm for recognizing the characters on a number plate. They used edge image of the number plate as features.
Chapter 3

Hypothesizing the Speed Sign Regions in the Input Traffic Image

The U.S. speed-limit signs are rectangular in shape with black characters printed on a white background. It has a black thick boundary separating the sign board from the rest of the image. Initially the input image is converted into grayscale image. A sample grayscale traffic scene is shown in the figure 3-1.

![Sample traffic scene](image)

Figure 3-1 Sample traffic scene

To extract the speed-limit information from the sign board present in the input image, first the sign board has to be located in the input image. The first phase of our algorithm tries to detect the speed-limit signs present in the input image. Here the detection phase
does not require high accuracy, because the recognition phase is able to recognize the true and false sign board regions. The first phase of the algorithm hypothesizes the speed-limit sign locations in the input image using the appearance based features of the sign board. First the input image divided into a set of regions, using connected component labeling technique adapted for grayscale images, and then each region parameters are calculated to choose the possible speed sign regions.

3.1. Introduction to Connected Component Labeling

Connected component labeling (CCL) is used to label the connected regions in the input image. It labels each pixel by studying its neighborhood pixels’ labels and deciding which pixel is more appropriate for the center pixel. We have quite a few algorithms existing in the literature to perform connected component labeling technique. Different criteria can be used to classify these algorithms. Some algorithms note the label equivalence and resolve them while some other algorithms label the image without the necessity of resolving the label equivalence. The algorithms that consider label equivalence can be divided based on the scan method they use. In multi-scan algorithms the image is scanned multiple number of times in forward and backward raster directions, interspersed with label equivalence resolution. In two pass algorithms the labels assigned in the first scan are resolved and updated in the second scan. In single scan algorithms the label equivalence is resolved and updated in parallel with the assigning labels. The algorithms that do not consider resolving the label equivalence have their own kinds of methods to avoid the analysis of redundant labels. Under this category multi scan algorithms go through the image in forward and backward raster directions alternately.
until there will be no changes in the assigned label values. In contour tracing algorithms it is achieved by tracing the contours of the objects.

The multi scan algorithms require more computations compared to the two pass algorithms. It is observed that a two pass algorithm is more appropriate for our application. We have chosen the basic two pass algorithm which employs label resolution and modified it to adapt it for the grayscale images. The original two-pass connected component labeling algorithm is explained in the next section.

3.2. Original Two-pass Connected Component Labeling Algorithm

Rosenfeld and Pfaltz [34] had proposed this algorithm to label the connected components in binary images. It comprises of three steps in it. This algorithm scans the image from left to right and top to bottom.

1. In the first scan each pixel in the image is labeled by studying its neighborhood pixels’ labels and deciding which label is more appropriate for the center pixel. In this step the label equivalence information is also noted.
2. In the second step the label equivalence is resolved using an equivalence matrix.
3. In the third step the image is scanned for the second time to update the resolved labels.

At each pixel a 4-neighbor forward raster scan mask such as shown in the figure 3-1 is chosen. The current pixel is positioned at “e” and the pixels at the remaining positions of the mask are its neighbor pixels. In binary images ‘1’ represents an object pixel and ‘0’ represents a background pixel.
The Rosenfeld and Pfaltz’s algorithm is explained as follows: if the current pixel value is zero then it is not changed and the mask is moved to the next pixel. If the current pixel value is one and all its neighbor pixels are zeros then a new label is assigned to the current pixel. The new label is obtained by increasing the maximum label value in the labeled image by one. If the current pixel value is one and there is only one neighbor pixel with a non-zero label value, then its label is assigned to the current pixel. If the current pixel value is one and there are two or more neighbor pixels with non-zero label values then the minimum label value among them is assigned to the current pixel and these non-zero neighbor labels are noted as equivalent and stored in a matrix. These steps are explained using the following equations. The labels assigned in the first scan are termed as provisional labels.

\[
g(x, y) = \begin{cases} 
0, & \text{if } b(x, y) = 0, \\
m, (m = m + 1) & \text{if } b(x, y) = 1 \text{ and } \forall \{i, j \in \mathcal{N}\} g(i, j) = 0, \\
g_{\min}(x, y) & \text{otherwise,}
\end{cases}
\]  
\[
g_{\min}(x, y) = \min\left[\{g(i, j)|i, j \in \mathcal{N} \text{ and } b(i, j) = 1\}\right]
\]  

Note, \(b(x, y)\) represents the intensity value of the input image “b” at location \((x, y)\), “g” represents the labeled image and \(g(x, y)\) indicates the label value assigned to the pixel at \((x, y)\), ‘N’ is the 4-neighborhood of the center pixel and ‘m’ represents the new label value.

In the second step the equivalence matrix obtained in the previous step is processed to resolve the label equivalence and remove label redundancies. In the final step the
image is scanned for the second time to update the label values with their equivalent classes. This algorithm produces the labeled regions in the given input image.

3.3. Modified Connected Component Labeling Algorithm for Grayscale Images

Unlike binary images grayscale images have intensity values spread over a range. The pixels belonging to a single object need not possess equal intensity values. The pixels of different objects or regions are distinguished by their connectivity and intensity values. The pixels belonging to the white region of the speed sign are connected and have intensity values approximately equal to each other. The thick black boundary of the sign board separates this region from the rest of the image. These features of the sign board make us use CCL algorithm for detecting this region in the input image. So in order to adapt the CCL technique for grayscale images few changes are made in its original algorithm. The steps involved in the modified algorithm are discussed below.

In a binary image two adjacent pixels are considered as connected if their values are equal otherwise they are considered as disconnected. In a grayscale image two adjacent pixels can be considered as connected if their intensity values are closer to each other. The difference between their intensity values is calculated and if it falls below a threshold value then the two pixels are considered as connected. The modified CCL algorithm also consists of three phases in it, scanning phase, label resolution phase and label updating phase. The same forward raster scan mask is used to consider the 4-neighborhood in this algorithm. At each pixel the mask is applied and the connectivity condition, explained before, is checked between the center pixel and each of the 4-neighbor pixels. If none of the 4-neighbor pixels satisfy the connectivity condition, then a
new label is assigned to the center pixel. If only one of the 4-neighbors satisfy the connectivity condition, then its label is assigned to the center pixel. If two or more of the 4-neighbor pixels satisfy the condition, then minimum label among them is assigned to the center pixel and all these neighbor labels are considered as equivalent and stored in a one dimensional array. Following equations explain these steps.

\[
g(x, y) = \begin{cases} 
    m, (m = m + 1) & \text{if } \forall \{i, j \in N\} \text{abs}(b(x, y) - b(i, j)) > T \\
    g_{\text{min}}(x, y) & \text{otherwise}, 
\end{cases} 
\]  

(3.3) 

\[
g_{\text{min}}(x, y) = \min\{\left\{g(i, j)|i, j \in N \& \text{abs}(b(x, y) - b(i, j)) \leq T\}\} 
\]  

(3.4)

3.3.1. Resolving Label Equivalence using Union-find Algorithm

In the second step the obtained label equivalence information is processed to resolve the label equivalent relations and to remove the redundant labels. We employed a union-find algorithm for this purpose. Union-find algorithm performs two operations, which are:

- Find: it determines the class of a particular element in the array.
- Union: it merges two classes into one class if they are equivalent.

The label equivalence information obtained during the first scan is stored in a one dimensional array such as shown in the figure 3-2. The indices of the array represent the provisional labels assigned in the first scan. The value at each index gives its equivalent label.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1(i)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 3-3 Label equivalent matrix
Where ‘i’ represents the index value and ‘L1’ is the equivalent matrix. In the above example the redundant labels are (3, 2), (4, 3), (6, 4) and (8, 6). This label equivalence is resolved using the following algorithm: the value at each index in the equivalent matrix is checked. If the value at an index and the index are equal then the value at the index is unchanged. If the value at the index and the index are not equal then the value at the index is replaced with its equivalent value. The resolved label matrix is shown in the figure 3-3.

```
   i  1  2  3  4  5  6  7  8
L2(i) [ 1  2  2  2  5  2  7  2 ]
```

Figure 3- 4 Resolved label equivalent matrix

In the final step of the CCL algorithm the image is scanned again and the redundant labels are replaced with their respective equivalent labels from the resolved label matrix. This gives the labeled regions in the input image. In the next step the labeled image is used to calculate the features of the regions and compare them with the speed-limit sign features.

3.4. Hypothesizing the Speed Sign Regions Using Region Parameters

After labeling the regions in the input traffic image, each region parameters are calculated and used to hypothesize the speed-limit sign regions in the image. A regular U.S. speed-limit sign has an aspect ratio of 0.8 or 0.8333, but this value would vary slightly when calculated in the captured traffic scenes. This variation is due to different perceptions, tilted sign boards and deformations.
We formed two conditions, using the characteristics of the regions, to hypothesize the speed-limit sign regions in the input image. First the size of the region is used as one condition to ignore very small and very large regions. The regions, whose sizes fall in the range of 1000 to 10,000 pixels, in a normal resolution image are chosen. Next the aspect ratio of a region is used as the second condition to finalize the hypothesis. Aspect ratio is defined as the ratio of width to height. The aspect ratios of the regions which satisfied the first condition are calculated next, using the following equation.

\[
\text{Aspect ratio of a region} = \frac{\text{Ending column no.} - \text{starting column no.}}{\text{Ending row no.} - \text{starting row no.}} \quad (3.5)
\]

The regions whose aspect ratios fall in the range of 0.5 to 1 are finally hypothesized as the speed-limit signs. These two conditions are found to be sufficient in generating good hypotheses. These hypothesized regions are further processed in the next phase to recognize them.
Chapter 4

Recognizing the Hypothesized Regions

This chapter discusses the recognition phase of the algorithm. The first phase of the algorithm detects the regions that have similar features as a speed-limit sign board. The size and the aspect ratio of the region are the two parameters used to hypothesize the speed sign regions in the input image. To detect the speed-limit sign board accurately its informative part, which are characters, should also be identified. Initially we don’t know the exact location of the sign board in the input image so, instead of performing the recognition for the entire image, by hypothesizing the sign board locations in the image, the recognition is performed only for these regions, thus reducing the computation time required. This chapter provides few introductory points about object recognition and then discusses the recognition method employed in our algorithm.

4.1. Introduction to Object Recognition

Object recognition is one of the most important topics of digital image processing. To recognize an object is to identify it or name it or specify its class or type. The applications of object recognition are spread in many fields like, medical imaging, industrial applications, remote sensing, sorting and categorizing, image retrieval and
object detection. Depending on the requirements, constraints and nature of the application different algorithms are proposed for object recognition. These algorithms can be categorized based on different criteria such as, type of the features used to describe the object or the approach of the recognition scheme or the matching strategy used.

Different algorithms utilized different features of the objects to perform object recognition. Based on object representation these algorithms can be broadly divided into two types.

i. Shape based methods.

ii. Appearance based methods.

Shape based algorithms utilize the geometrical features of the objects like corners, edges, blobs and ridges and appearance based algorithms use the features calculated from the regions covered by the objects.

Based on the recognition schemes used, object recognition algorithms can be categorized into three types.

i. Invariant properties methods.

ii. Parts decomposition methods.

iii. Alignment methods.

The invariant properties of the objects under different object transformations are recognized in the first kind of algorithms. These methods employ techniques like feature spaces, clustering and separation. The second kind of methods relies on decomposing the object into its constituent parts. Each part is recognized individually to identify the object on the whole. This approach leads to algorithms like symbolic structural descriptions, syntactic pattern recognition and feature hierarchies. In the alignment methods an object
is recognized by measuring the transformations the objet has gone under and comparing the object with the available object models in measured transformations. It is not necessary that a given recognition algorithm should belong to only one these kinds, it could combine the features from different methods. Our proposed algorithm is one such kind; it aims at dividing the speed-limit sign into its constituent parts, which are characters printed on the sign board, and identifying them using their invariant properties.

4.2. Steps Involved in the Proposed Recognition Algorithm

The first phase of the algorithm extracted the possible speed-limit sign regions by analyzing the region characteristics in the input image. The second phase of the algorithm known as recognition phase aims at classifying each extracted region as a speed-limit sign board region or not and extracting the assigned speed limit information from it if it is classified as a speed-limit sign. For this the inner parts of the extracted regions need to be analyzed. The inner part of the speed-limit sign contains the printed characters. The upper half of the speed-limit sign contains the alphabets ‘S’, ‘P’, ‘E’, ‘E’, ‘D’, ‘L’, ‘I’, ‘M’, ‘I’, ‘T’ and the lower half contains the assigned speed limit printed in numerical value. Hence by analyzing the structural components of the extracted regions from the first phase of the algorithm, the sign board can be identified. The steps involved in the recognition phase are given as,

1) Pre-processing
2) Feature extraction and
3) Classification
4.2.1. Pre-processing

This is the first step in the recognition phase. The extracted regions in the first phase possess similar aspect ratio as the speed-limit sign. Each hypothesized region is processed one after the other. First the region is resized to a common size to utilize the spatial information in the classification stage. Then it is normalized, converted into a black and white image and then it is inverted to convert the white background of the speed-limit sign into black and the black characters of it into white. Then each individual component of the resulting image is labeled using the following MATLAB function,

\[ L = bwlabel(bw, n) \]  \hspace{1cm} (4.1)

Where, ‘bw’ represents the black and white image, ‘L’ is the matrix containing labels and ‘n’ is the variable specifying the type of the connectivity of the objects in the input image. The value of ‘n’ can be 4 or 8 and when it is omitted the function assumes an 8-connectivity. The function assigns unique label values to each of the connected objects in the input image.

![Figure 4-1 (a) Normalized image (b) Inverted black and white image (c) Labeled image](image)

The Figure 4-1 (c) shows the labeled image after pre-processing the speed-limit sign board extracted during the detection phase. Each component in the image is chosen by its label value and processed for recognition.
4.2.2. Feature Extraction from Hu’s Moments

The next step in the recognition phase is extracting useful and robust features of the inner parts of the hypothesized region. The prominent or distinct properties, qualities or characteristics of an object are called its features. For example corners, edges, texture, size, length, orientation and blobs are some of the features used in the image processing applications. Feature extraction refers to calculating these characteristics of the object that can be used to define it and represent it with less data than the original data. The success of an image processing algorithm depends on the quality of the features used in it. If the nature of the features calculated under different normal and noisy conditions is consistent then the algorithm would be successful, otherwise the algorithm would not be successful. Different applications encounter different kinds of situations like, occlusions, illumination variations, transformed images, background clutter and poor contrast etc. To design a successful algorithm one needs to tackle these kinds of situations by focusing on extracting application-dependent features. For example, speed-limit signs tend to tilt, rotate, are viewed at different angles and partially occluded sometimes. So the features extracted under these kinds of noisy conditions should be able to capture the characteristics of the object, which would in turn help for a successful algorithm.

Each hypothesized region in the detection phase of our algorithm is broken into its constituent parts in the previous step. In this (feature extraction) step, each part of the region is chosen to calculate its features. If the extracted region is a speed-limit sign then the characters on it are its constituent parts. The recognition of optical characters is one of the popular applications of the image processing. In the case of speed-limit sign image the recognition of only the specific characters written on it is required, but not all the
characters. The general obstacles faced in this process are the rotated, rescaled and tilted characters. Hence the extracted features should be robust enough to tackle these conditions.

In our algorithm features are extracted from the Hu's moments. Hu’s moments, also called as geometric moment invariants, are introduced by Hu [18]. They were used in the identification of aircraft, radar images, matching of optical images and texture classification. They have the following advantages,

- they produce a set of invariant features
- they are able to capture the global and geometrical features of the image

The Hu’s moments are derived from the normalized central moments, which in turn are derived from the raw moments of the image. The raw moments of a digital image ‘f(x, y)’ of the order (p+q) are represented by the symbol, ‘m_{pq}’ and are given as follows,

\[
m_{pq} = \sum_x \sum_y f(x, y) \ x^p \ y^q
\]  

These raw moments are used to derive simple image properties like area and center of mass. The (p+q)th order central moments of the image ‘f(x, y)’ are given by the following equation.

\[
\mu_{pq} = \sum_x \sum_y f(x, y) (x - \bar{x})^p (y - \bar{y})^q
\]  

Where,

\[
\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}}
\]

Image orientation information can be calculated from the central moments. These central moments are made invariant to translation and variations in scale by multiplying them with a factor derived from the (00)th central moment.
\[ \eta_{pq} = \frac{\mu_{pq}}{\mu_{00}} \]  \hspace{1cm} (4.4)

Where,

\[ \gamma = \frac{(p+q)}{2} + 1, \text{ for } (p+q) \geq 2. \]

Hu combined different orders of these normalized central moments to produce a set of seven moments which are invariant to translation, scale and even rotation. These seven moments are given as below,

\[ \varphi_1 = \eta_{20} + \eta_{02} \]
\[ \varphi_2 = (\eta_{20} - \eta_{02})^2 + 4 \eta_{11}^2 \]
\[ \varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \]
\[ \varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \]
\[ \varphi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \]
\[ \varphi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \]
\[ \varphi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] \]  \hspace{1cm} (4.5)

Of the seven Hu’s moments, when calculated them for different characters, it is observed that the last three moments are very minute and exhibit very little difference among different patterns of the characters. Keeping these observations in mind we have formed a feature vector that comprises of the first four moments, as shown below.

\[ [M] = [\log(\varphi_1 \times 10) \quad \log(\varphi_2 \times 10^2) \quad \log(\varphi_3 \times 10^4) \quad \log(\varphi_4 \times 10^4)] \]  \hspace{1cm} (4.6)

The moments are multiplied by appropriate factors to bring equilibrium in to the feature vector. By taking the logarithmic values the dynamic range of the feature vector is reduced. These four features are used in the next step for classifying the hypothesized
regions and extracting the speed information from it if it is classified as a speed-limit sign.

4.2.3. Region Classification and Speed Information Extraction

The aim of this step is twofold. First the hypothesis generated in the first phase of the algorithm is verified by classifying the region as speed-limit sign or not. Second the information regarding the assigned speed limit available from the region is extracted. Both these parts require the recognition of the characters that present on a speed-limit sign board. The characters on the upper half of the sign board will confirm its identity, while the lower half components will specify the assigned speed limit. The features of these characters are matched with the features of the components of the detected regions for recognition. The components of the detected region are termed as objects and the characters present on a speed-limit sign board are termed as templates. The classification of the region is performed as follows.

4.2.3.1. Classification by Matching Features and Locations

The classification of the regions is performed by matching the locations and features of the objects with that of templates. By observing the locations of the alphabets in the speed-limit sign a location range is assigned for each alphabet for its beginning row and column. As mentioned already the hypothesized region is resized to the size of the database speed-limit sign image. If the beginning row and column of an object in the region are in the range of a template alphabet’s assigned location range then the object and the template are considered as location matched. Next the features of the location matched object and template are compared by measuring the correlation coefficient
between their feature vectors. The feature vectors for the characters of the speed-limit sign are already calculated and stored in the database. These templates feature vectors are represented by the variable ‘\(M_t\)’ and the feature vectors of the objects are represented by ‘\(M_o\)’. The correlation coefficient ‘\(C\)’ between the two feature vectors of the object and the template is calculated using the following equation.

\[
C = \frac{\sum_{i=1}^{4} (M_o(i) - \bar{M}_o)(M_t(i) - \bar{M}_t)}{\sqrt{(\sum_{i=1}^{4} (M_o(i) - \bar{M}_o)^2)(\sum_{i=1}^{4} (M_t(i) - \bar{M}_t)^2)}}
\] (4.7)

Where ‘\(\bar{M}_o\)’ and ‘\(\bar{M}_t\)’ represent the mean values of the feature vectors. If the value of the correlation coefficient between the object and the template is above a chosen threshold, then they are considered as similar otherwise it is not considered as a match. This process is repeated for all the objects of the detected region and the number of matches found is noted. If the number of matches found are sufficient enough, then the region is classified as a speed-limit sign otherwise it is rejected. A U.S. speed-limit sign contains 10 alphabets in its upper half. If the number of matches found is greater than or equal to 6 then the region is classified as a speed-limit sign. The regions classified as speed-limit signs are further processed to extract the information regarding the assigned speed limit.

### 4.2.3.2. Speed Limit Information Extraction Using Neural Networks

The assigned speed limit is printed on the lower half of the sign board. The speed limits in U.S. range from 10 mph to 85 mph and they differ by multiples of 5. So the extraction of the speed limit information from the sign board requires the recognition of the integers from 0 to 8. A multilayer feed forward neural network (FFNN) with a back propagation learning algorithm is used for this purpose.
A multilayer FFNN consists of one input layer, one output layer and at least one hidden layer in between them. Each layer consists of a number of neurons, which are analogous to the neurons in the brain. The neurons in the input layer are connected to the neurons in the hidden layer and the neurons in the hidden layer are connected to the neurons in the output layer as shown in the figure 4-2. A weight is associated with each neuron-neuron connection and the network learns by repeatedly adjusting these weights. Each neuron receives one or more input signals but always produces a single output signal. The input to the network is given to the neurons in the input layer and the output is obtained from the output layer neurons.

The network is trained using the back propagation learning algorithm. The steps involved in it are explained as follows:
Step 1: Initialization

The weights and the threshold levels of the network are chosen randomly from a small range of \((-\frac{2.4}{F_i}, +\frac{2.4}{F_i})\).

Where ‘\(F_i\)’ represents the number of inputs to neuron i.

Step 2: Activation

The inputs \(M_1, M_2, \ldots, M_n\) from the training set and the corresponding outputs \(y_{d,1}, y_{d,2}, \ldots, y_{d,n}\) are applied to the neurons in the input and output layers respectively.

a) The actual outputs of the hidden layer neurons are calculated using the following equation:

\[
y_j(p) = \text{sigmoid} \left[ \sum_{i=1}^{n} M_i(p) \times w_{ij}(p) - \theta_j \right] \tag{4.8}
\]

Where ‘\(p\)’ is the iteration value, ‘\(\theta_j\)’ is the threshold level at neuron j, ‘\(w_{ij}\)’ is the weight of the link connecting neurons i and j and the sigmoid function is given by the following equation,

\[
y_{\text{sigmoid}} = \frac{1}{1+e^{-X}} \tag{4.9}
\]

Where ‘\(X\)’ represents the input value to the sigmoid function.

b) And the actual outputs at the output layer are calculated as follows,

\[
y_k(p) = \text{sigmoid} \left[ \sum_{j=1}^{m} y_j(p) \times w_{jk}(p) - \theta_k \right] \tag{4.10}
\]

Where \(m\) represents the number of neurons in the hidden layer and \(\theta_k\) is the threshold level at neuron k in the output layer.

Step 3: Adjusting the weights

After feeding the input signals in the forward direction in the network to calculate the outputs at neurons in hidden layer and output layer, now the error values are calculated at output layer and are propagated in the backward direction.
The error and error gradient values for the neurons in the output layer are calculated:

\[ e_k(p) = y_{d,k}(p) - y_k(p) \]  
\[ \delta_k(p) = y_k(p) \times [1-y_k(p)] \times e_k(p) \]

Where \( e_k \) is the error at neuron \( k \) and \( \delta_k \) is the error gradient value.

The corrections for the weights are calculated as,

\[ \Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p) \]

Where ‘\( \alpha \)’ is the learning rate. The weight corrections are added to the weights in the current iteration to calculate the new weights.

\[ w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \]

b) The error gradient value for hidden layer neurons are calculated,

\[ \delta_j(p) = y_j(p) \times [1-y_j(p)] \times \sum_{k=1}^{m} \delta_k(p) \times w_{jk}(p) \]

And the weight corrections are calculated as,

\[ \Delta w_{ij}(p) = \alpha \times M_i(p) \times \delta_j(p) \]

The weights of the connections are updated as,

\[ w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \]

Step 4: Iteration

The value of the iteration \( p \) is increased by one and the steps 2, 3 and 4 are repeated until the chosen error criterion is satisfied. Some methods fix number of iterations while others calculate the mean square error value to terminate the process.
After training the network and adjusting the weights, the feature vectors of the inner parts of the detected region are fed as inputs to the network to recognize them. The output is collected from the output neurons, which gives the assigned speed limit.
Chapter 5

Test Results and Analysis

The proposed algorithm is experimented on different traffic scenes and the results obtained are presented in this chapter. The traffic scene, which appears in front of a vehicle, is captured by the camera mounted on its rear view mirror, and processed using the proposed algorithm in mat lab. There are two phases in the algorithm: sign board detection phase, which detects the possible speed-limit signs in the input image and sign board recognition phase, which verifies the hypotheses generated in the first phase and extracts the information from it by recognizing the inner components of the regions. Each phase has a number of steps involved in it. The results obtained after each important step are presented in this chapter in the order they are produced.

The detection phase employs the proposed modified connected component labeling technique, which adapts to the grayscale images. A threshold level of 35 is used to check the connectivity condition, performed in the CCL algorithm, which is observed to be the efficient value in labeling the speed-limit sign distinctly. The neural network used in the recognition phase consists of 4 neurons in input layer, 7 neurons in hidden layer and 9 neurons in output layer. This network is trained using the Hu’s moments of the characters written on a speed-limit sign board.
5.1. Experiment 1

5.1.1. Detection Phase

The image captured by the camera is converted into a grayscale image, which is shown in the figure below.

![Original image](image1.jpg)

Figure 5-1 Original image

The modified connected component labeling technique is used to label this grayscale image. The algorithm labels the connected regions distinctly.
After labeling the input grayscale image, the regions in the image are analyzed using the region characteristics. We have used the size and the aspect ratio of the regions to hypothesize the locations of the speed-limit sign in the input image.

**Condition 1: Region size**

The input image contains a lot of small objects like the characters on sign boards, number plates and few very big regions. We have chosen the regions with sizes in the range of 1000 to 10,000. This condition would be helpful to ignore the very small and very big regions. The size of the regions is calculated by counting the number of pixels in them. The variation of the region sizes in the labeled image (figure 5-2) is shown in figure 5-3, using the histogram of the labeled image. The horizontal axis represents the label value of the region and the vertical axis represents the size of the region.
Condition 2: Aspect ratio

As mentioned in the chapter 3, we have chosen the regions whose aspect ratio falls in the range of 0.58 to 1. The table below lists few regions and their sizes and aspect ratios.

Table 5.1  Examples of few regions’ characteristics from the figure 5- 2

<table>
<thead>
<tr>
<th>Region label value</th>
<th>Size of the region</th>
<th>Aspect ratio of the region</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1908</td>
<td>2.4</td>
</tr>
<tr>
<td>42</td>
<td>1830</td>
<td>0.807</td>
</tr>
<tr>
<td>168</td>
<td>2175</td>
<td>3.1</td>
</tr>
</tbody>
</table>

From the above table we can see that the region with label value 42 satisfies both our conditions. This region is hypothesized as the speed-limit sign region and the resulted image is shown in the figure below.
5.1.2. Recognition Phase

5.1.2.1. Pre-processing

The beginning and ending coordinates of the hypothesized region are noted and this part of the original image is stored in a separate matrix. This image is then resized to common size, converted into a black and white image, inverted the colors and labeled the individual components in it. This entire pre-process is shown in the figure below.

![Figure 5-4 Generated hypothesis](image)

Figure 5-4 Generated hypothesis

5.1.2. Recognition Phase

5.1.2.1. Pre-processing

The beginning and ending coordinates of the hypothesized region are noted and this part of the original image is stored in a separate matrix. This image is then resized to common size, converted into a black and white image, inverted the colors and labeled the individual components in it. This entire pre-process is shown in the figure below.

![Figure 5-5](image)

Figure 5-5 (a)Detected region (b)Inverted black and white image (c)Labeled image

The labeled image is used to select the individual components of the region and the black and white image is used to calculate their features.
5.1.2.2. Feature extraction

In this step each component of the hypothesized region is chosen separately and its feature vector is formed by calculating its Hu’s moments.

\[
[M] = [\log(\Theta_1 * 10) \log(\Theta_2 * 10^2) \log(\Theta_3 * 10^4) \log(\Theta_4 * 10^4)]
\]

5.1.2.3. Classification

In this step the hypothesis generated in the detection phase is verified by recognizing the upper half components of the region. The classification method uses the geometrical features derived from Hu’s moments and the spatial information of the individual components of the region. The region is classified as a speed-limit sign if there are at least 6 recognized characters at specified locations in the region. The results of the classification step are shown in the table 5-2.

![Speed Limit 65](image)

Figure 5-6 Analyzing the upper half of the detected region

<table>
<thead>
<tr>
<th>No. of components in the region</th>
<th>No. of recognized characters</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>Speed-limit sign</td>
</tr>
</tbody>
</table>
5.1.2.4. Speed Information Extraction

The region is further processed only if the region is classified as a speed-limit sign. The assigned speed limit is printed on the lower half of the sign board.

![Image of speed limit sign](image)

Figure 5-7 Analyzing the lower half of the region

We have trained a neural network with 4 input layer neurons, 7 hidden layer neurons and 9 output layer neurons for recognizing the numbers from 0 to 8. The network is trained with a back propagation learning algorithm. The feature vectors of the speed-limit sign characters are fed as inputs to the network and it is trained for 10,000 iterations. The image for each speed limit is already available in the data base and the recognized speed limit’s sign board is displayed as output.

![Image of recognized speed limit](image)

Figure 5-8 Recognized speed limit
5.2. Experiment 2

![Original image](image1)

Figure 5- 9 Original image

![Labeled image](image2)

Figure 5- 10 Labeled image
Figure 5-11 Histogram of the labeled image showing the sizes of the regions

Table 5.3  Some of the regions parameters from the labeled image

<table>
<thead>
<tr>
<th>Region label value</th>
<th>Size of the region</th>
<th>Aspect ratio of the region</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1123</td>
<td>0.729</td>
</tr>
<tr>
<td>182</td>
<td>1162</td>
<td>0.961</td>
</tr>
</tbody>
</table>

Figure 5-12 Generated hypothesis
Table 5.4 Classifying the hypothesized regions in the figure 5-12

<table>
<thead>
<tr>
<th>Hypothesis No.</th>
<th>No. of upper half components</th>
<th>No. of recognized characters</th>
<th>Classification result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Not a speed-limit sign</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>Speed-limit sign</td>
</tr>
</tbody>
</table>

The hypothesis 1 is rejected in the classification step and the hypothesis 2 is classified as speed-limit sign and its lower half components are recognized using the trained neural networks. And the recognized speed limit sign board is displayed in the figure below.

![Figure 5-13 Recognized speed-limit sign](image)
5.3. Experiment 3

Figure 5-14 Original image
Figure 5-15 Labeled image
Figure 5-16 Histogram of the labeled image

Table 5.5 Region parameters from the labeled image

<table>
<thead>
<tr>
<th>Region label value</th>
<th>Size of the region</th>
<th>Aspect ratio of the region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1711</td>
<td>1272</td>
<td>0.771</td>
</tr>
</tbody>
</table>

By analyzing the region parameters from the labeled image we obtained only one region that satisfies the two conditions.
Table 5.6 Classifying the hypothesized region in the figure 5-17

<table>
<thead>
<tr>
<th>No. of upper half components</th>
<th>No. of recognized characters</th>
<th>Classification result</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>Speed-limit sign</td>
</tr>
</tbody>
</table>

In the next step the lower half components of the region are recognized using the trained neural network and the recognized speed limit is displayed as the sign board as shown in the figure below.
5.4. Experiment 4

Figure 5-18 Recognized speed-limit sign

Figure 5-19 Original image

Figure 5-20 Labeled image
Figure 5-21 Histogram of the labeled image

Table 5.7 Region parameters from the image 5-21

<table>
<thead>
<tr>
<th>Region label value</th>
<th>Size of the region</th>
<th>Aspect ratio of the region</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>3302</td>
<td>0.887</td>
</tr>
<tr>
<td>23</td>
<td>1981</td>
<td>0.456</td>
</tr>
</tbody>
</table>

Using the region parameters calculated from the labeled image the possible speed-limit sign locations are shown in the figure below.
Table 5.8 Classifying the hypothesized region in the figure 5-22

<table>
<thead>
<tr>
<th>No. of upper half components</th>
<th>No. of recognized characters</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9</td>
<td>Speed-limit sign</td>
</tr>
</tbody>
</table>

Figure 5-23 Recognized sign board
5.5. Experiment 5

Figure 5-24 Original image
Figure 5-26 Histogram of the labeled image

Table 5.9 Region parameters from the labeled image in figure 5-26

<table>
<thead>
<tr>
<th>Region label value</th>
<th>Size of the region</th>
<th>Aspect ratio of the region</th>
</tr>
</thead>
<tbody>
<tr>
<td>679</td>
<td>1594</td>
<td>0.8363</td>
</tr>
<tr>
<td>1108</td>
<td>7934</td>
<td>0.6012</td>
</tr>
</tbody>
</table>
Figure 5-27 Generated hypotheses

Table 5.10 Classifying the hypothesized regions in the figure 5-27

<table>
<thead>
<tr>
<th>Hypothesis no.</th>
<th>No. of components in the region</th>
<th>No. of recognized characters</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Not a speed-limit sign</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>7</td>
<td>Speed-limit sign</td>
</tr>
</tbody>
</table>
Figure 5-28 Recognized sign board
Chapter 6

Conclusions and Future Work

This thesis presents a new method for automatically detecting the speed-limit signs positioned on either side of a road. We have modified the original connected component labeling technique to adapt to the grayscale traffic images. A different connectivity rule is designed to apply it on the grayscale images. We used this technique to label the individual regions of the input image. The parameters of the labeled regions are calculated and matched with the parameters of a speed-limit sign image to hypothesize the locations of the speed-limit signs in the input image. This forms the detection phase of our algorithm. In the next phase, called as recognition phase, the individual components of the regions are labeled and their invariant features are calculated from Hu’s moments. The components in the upper half of the region are first analyzed to identify the region. If the region contains the specific alphabets that present on a speed-limit sign, then the region is classified as a speed-limit sign and processed further, otherwise it is rejected. If the region is classified as a speed-limit sign, then its lower half components which specify the assigned speed limit are identified using a feed forward neural network, trained with a back propagation learning algorithm.
Traffic sign boards are often rotated, tilted, partially occluded and are viewed at different angles. Under these noisy conditions appearance based features are more robust than the shape based features. Hence we have utilized the region characteristics instead of the geometrical features. The proposed recognition method includes features from two different kinds of recognition methods; they are decomposing the region into its structural parts and using the invariant features for the recognition scheme. This allows achieving more accurate results.

The connected component labeling algorithm requires more computations when applied on grayscale images, as it has to label every pixel unlike in binary images. The number of computations can be probably reduced by defining the domain of operation for the algorithm. In the recognition phase, we have employed two different methods for classifying the region and extracting the information, although both do the same job. It is because the neural network is not able to reject the unknown patterns. If the neural network is equipped with such a mechanism then it would further simplify the method.
References


