A CAMERA BASED SPEED LIMIT SIGN RECOGNITION SYSTEM

Jim Torresen*, Jorgen W. Bakke and Yizhuang Yang
Department of Informatics
University of Oslo
P.O. Box 1080 Blindern
N-0316 Oslo, Norway
E-mail: jimtoer@ifi.uio.no
Web: http://home.ifi.uio.no/~jimtoer

ABSTRACT

To improve traffic safety, an automatic traffic sign detection system would be important to assist the driver. In this paper, an approach for detecting Norwegian speed limit signs is proposed. It consists of three major steps: Color-based filtering, locating sign(s) in an image and detection of numbers on the sign. About 91% correct recognition is achieved for a selection of 198 images. Ongoing work is focused on video input analysis.

KEYWORDS

Sign Recognition, Speed Limit, Image Processing and Real-Time Processing.

1 INTRODUCTION

A system for automatic recognition of traffic signs would probably be important for traffic safety in the future. Such systems could assist drivers on signs they did not notice before passing them. Specifically speed limit sign recognition – studied in this paper, could inform drivers about the present speed limit as well as giving an alert if a car is driven faster than the speed limit. In the future, autonomous vehicles would probably have to be controlled by automatic road sign recognition.

Recognizing road images have been studied for a long time. The first known attempt for making a real-time system was by Akatsuka and Imai [1]. Many techniques have been proposed since then.

The standard technique for detecting and recognizing road signs consists of three steps [2]. First, color segmentation or color thresholding is applied to emphasize possible signs in an image. Thus, restricting the search area in the image. Second, template matching is applied for shape detection. Third, specific signs are detected using template matching or neural networks.

A color image is often transformed from the RGB (Red, Green, Blue) color space into the HSV (Hue, Saturation, Value) color space. Color segmentation then becomes easier (by applying it on the hue value only rather than the three RGB values). The hue value is invariant for the illumination as well. However, the hue is not suited for grey-level segmentation since it has a constant value along the "grey-level" axis. The value will be unstable near this axis too. Small perturbations in the RGB signals may cause large variations in the hue.

A traffic sign detection system consisting of four stages has been proposed by Wei et al [3]. First, a color-based filtering is applied to filter out image regions that have color characteristics similar to the color found in one of the signs known to the system. That is, the color input image is converted into a binary image using the filter. Second, the boundaries of the regions are smoothed by applying close and open morphological operations. Third, shape analysis is performed to evaluate the similarity between the region and a given shape. This includes major and minor axis as well as comparing area. Finally, the holes within the regions are analyzed. The system performs well for four different signs in different environments as well as at different orientation and scales.

Another system proposed by Hsu and Huang [4] is based on three steps for road sign detection followed by sign recognition based on Matching Pursuit (MP) filter. The detection locates regions with signs in the unknown input image by first using a priori knowledge about position, shape and color (HSV). Second, template matching is performed in the extracted regions to find road signs. Third, the road sign is extracted by further template matching and normalization.

This paper concerns detection of speed limit signs specifically. Earlier we have proposed algorithms for speed limited sign detection [5]. It has later been optimized for both better performance and speed [6]. Results from our experiments with sign number classification in evolvable hardware are given in [7]. In this paper we will continue by focusing on sequences of images.

We have found very little work on speed limit sign classification. There exists a system based on Global Position System (GPS) and digital road maps with speed limits included [8]. However, such a system depends on much external infrastructure. Further, problems like lack of updated speed limits on road maps question the reliability of such a system.

The next section introduces the Norwegian speed limit signs. This is followed by an outline of the proposed algorithm in Section 3 and results from the implementation in Section 4, respectively. Finally, Section 5 concludes the paper.

2 NORWEGIAN SPEED LIMIT SIGNS

Speed limit signs have features making them feasible for automatic detection. First, there is a limited number of signs to distinguish.

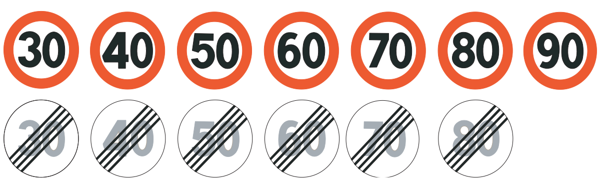

Figure 1: **Norwegian speed limit signs (standard).**

Figure 2: **An example of an input image.**

In Norway, the following speed limit signs – see Figure 1, are used: 30, 40, 50, 60, 70, 80 and 90 (100 are being tested at the moment). The speed is measured in kilometers per hour. A typical input image is shown in Figure 2. The outer circle of a sign is in *red* color. Further, there are signs to nullify 30 and 40 (set speed limit to 50), 50, 60 and 70 (set speed limit to 80) and 80 (set speed limit to 90). The nullify signs have not yet been considered by the algorithm outlined in this paper.

Second, there are rules (named a road grammar) for how signs can be placed along the road. After a "80" sign you will never find a "30" sign, but rather another "80" sign or a "60" sign. Third, the numbers on the signs are positioned vertically making the matching simpler. In curves only, the signs may be marginally tilted. These features make it promising to undertake speed limit sign detection with a very high rate of correct prediction.

Speed limit signs in several other countries use the same format (color and numbers) as in Norway. However, in some countries like in the US both the color and the numbers are different. Analyzing such signs would have to also consider the *second* digit since it can be either 0 or 5 (a speed limit could be e.g. 55).

Making a speed limit sign detection system commercially applicable, the safety is very important. If the detection system do not have a distinct best match, it should rather output that it "don't know" the speed at the moment. Thus, it is important that the system refrain from indicating a speed rather than indicating a *wrong* speed. The system presented in the next section is based on a novel heuristic developed through a large range of experiments.

3 NORWEGIAN SPEED LIMIT SIGNS DETECTION SYSTEM

The detection algorithm is divided into *three* parts: 1) Image filtering to emphasize the red parts of the sign(s), 2) Template matching to locate the sign(s) in an image and 3) Sign number recognition [6]. These will be presented in the following sections below.

3.1 IMAGE FILTERING

In this part, a specialized robust color filter is applied on the image to mark the *red circle* surrounding the speed limit numbers. It consist of two main parts. The first part – called Image filtering RWB (Red White Black), extracts the three colors red, white and black. The second part - called Red Reduction, reduces the number of red pixels in the image to minimize succeeding template matching processing. The detection algorithm has to use a broad definition of red and white colors since various intensities and saturations can occur in real images (depending on weather, day/night-light, etc.). On the other hand, only red color *on* traffic signs should be detected to minimize the number of starting points for the following template matching algorithm. Thus, detection of red color *outside* a traffic sign (like on buildings, cars, etc.) should be minimized.

The RGB color model was found useful for detection. RGB is also appropriate for hardware implementation since this will be the output from the camera. Converting into e.g. the HSV model would be computational expensive. The colors are defined (by experiments) as follows in the Image filtering RWB:

- A pixel is RED if: (R>77) AND (R-G>17) AND (R-B>17)
- A pixel is WHITE if: (R>108) AND (G>108) AND (B>108)
- A pixel is BLACK if: (R<122) AND (G<122) AND (B<122)

RED, WHITE and BLACK are symbolic names that will be used below. As one can see, the definition of the basic colors is broad to assure robustness. WHITE and BLACK are used mainly in different parts of the algorithm. Thus, there is a value overlap (108 to 122) between BLACK and WHITE. Good selectivity is achieved by combining these colors during pixel classification. The output from filtering the image in Figure 2 is given in Figure 3.

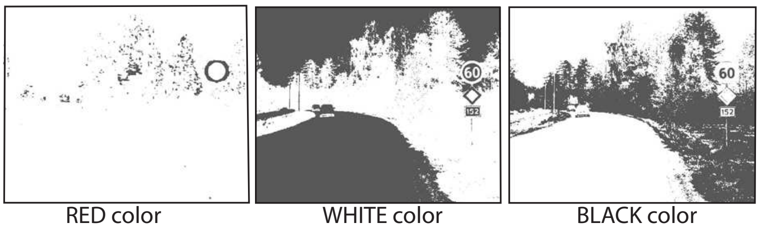

Figure 3: **The result of each of the three color filters. (Dark color marks pixels with the filtered color in each of the images.)**

The brightness varies a lot between the images. To be able to find signs in either light or dark images, the pixel values have to be made darker or lighter, respectively. This is undertaken if the average pixel values in an image is less than 125 (too dark image) or above 150 (too light image). The pixel values are to be modified by the difference between the average pixel value and the dark or light threshold values given above. Adjusting the value of each pixels in an image adds much computation time. Thus, the *thresholds* for the colors (RED, WHITE and BLACK) are modified instead. This results in the same recognition performance but less time is needed.
The following filtering algorithm is applied to reduce the number of red pixels in Red Reduction:

1. Find all 2x2 WHITE squares in the image (since there are white pixels *inside* the red circle on the sign).

2. Find (recursively) all RED neighbors of the located WHITE squares.

3. Apply template matching (described in the next section) to the RED pixels found.

This algorithm effectively limits the number of red pixels in the image to be further processed.

3.2 LOCATING SIGN BY TEMPLATE MATCHING

We locate signs in the image by searching for the *red circle* surrounding the numbers. The search is based on template matching. One reason for this is that template matching can be very effectively implemented in hardware. To limit the computation time, we have found that six templates are sufficient. This was based on analyzing sign sizes in images taken from a vehicle. The templates are of size 32x32, 38x38, 46x46, 52x52, 62x62 and 78x78 pixels and are used to detect the *position* of a speed limit sign. The templates are shown in Figure 4. We need a very high reliability of the system and small templates tend to produces incorrect classification of the numbers. The template matching algorithm is described in more detail in [6]. In addition to locating the sign, the algorithm tries to reject objects that are not signs and signs that are not speed limit signs. That is, to improve recognition at the same time as reducing the computation time.

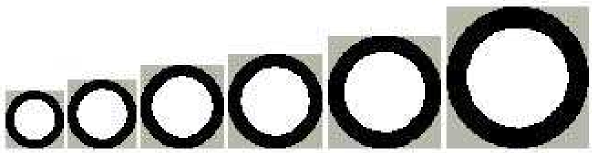

Figure 4: **Templates used for locating signs in an image.**

3.3 SIGN NUMBER RECOGNITION

The last part of the algorithm is to detect the speed limit *number* on a sign. This is conducted as follows:

1. Clean the space defined by the best template (remove RED and surrounding colors), but keep the numbers.

2. Find boundaries of numbers (width and height).

3. Work only with the first number (the second is always zero).

4. Create a 7 (rows) x 5 (columns) bit array of the given number (down-scaling): Set each bit of the array to 1 if there is more BLACK than WHITE pixels, else set the bit to 0.

5. Classify the bit array using a classifier system.

To classify the number given by the 7 x 5 bit array, we have got a high detection rate both with a feed-forward neural network trained by the back-propagation algorithm [6] as well as classification by evolvable hardware [7]. The experiments have so far been based on single images. Thus, future work would consist of implementing a real-time prototype system. This would be required to be able to verify the system in a real environment.

3.4 A SEQUENCE OF IMAGES

To reduce computation at the same time as improving the recognition performance, realistic experiments should be run on a video sequence of images. There are several useful properties in a sequence of images including the position and the size of a detected sign in the *previous* image. This information can be used to limit the search in an image when a sign is detected in the previous image as illustrated in Figure 5. In (b) only the light area is applied for image filtering and template matching. Further, only the *same* size or *larger* templates than the one giving best matching in the previous image need to be applied. This is due to the sign increasing in size as the vehicle is approaching it. The left-hand part of the image is included in the search since there may appear a sign on the opposite side of the road (of where the sign has been detected) and the sign earlier detected may temporarily be hidden by some obstacle. If a sign has been found in an image and not in the following one, the system should switch to do full image search.

The recognition will be improved by comparing the extracted numbers from a *series* of images and conclude on a new speed limit sign only if a confident number (varies with the speed of the vehicle) of images has given the *same* analysis results on speed limit numbers.

(a) Sign detected in image i (b) Reduced search in image i+1

Figure 5: **Search for a sign can be reduced when a sign is detected in the previous image.**

The limits for the processing in the next image is illustrated in Figure 6. Turns and uneven road may lead to the sign in the next image being positioned either higher or lower. The center of the search area is the *center* of the template having the best match in the previous image. The size of the area is set to be two times the *size* of this template. Our experiments so far indicate that this is sufficient for increased sign size in the next image as well as possible uneven road tilting the camera up or down.

We have done some preliminary experiments with sequences of images and have got promising results. This is based on a CMOS camera system provided by Micron. The image chip is targeted at automotive applications and should provide good image quality

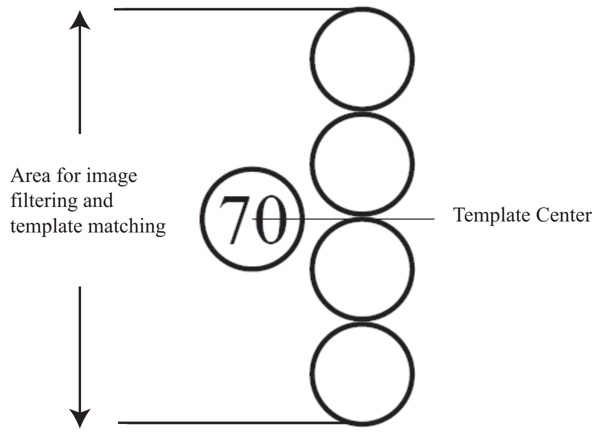

Figure 6: **Computation of limits for the next image to be processed.**

for different weather conditions. Further, a special shutter feature makes it possible to have all pixles sampled at the same time for the whole image. This avoids errors in the images that otherwise may occur by a fast moving camera.

4 PERFORMANCE RESULTS

4.1 SINGLE IMAGES

A database of 198 images from traffic situations were used in the experiments. 115 contained a speed limit sign and 83 contained other signs or no sign at all. Many of the images were in various ways "difficult" (different brightness on sign, faded color on sign etc). The results were as follows:

- Is there a speed limit sign? A speed limit sign was found (and correctly classified) in 100 of the 115 images (87%) containing a speed limit sign. In those not found, the system stated that a speed limit sign was not present in the image.
- 78 of the 83 images without a speed limit was correctly refused (94%). Thus, only five images were sent to the final sign number recognition. Two out of these were refused in the final stage by having an ambiguous/low output from the neural network. Those three matching a number could most probably be avoided by matching on the "0" number (not yet implemented).
- In summary, 180 out of the 198 images (90.9%) were correctly classified.

For *all* the 100 images that the system correctly detected a speed limit sign, the number on the sign was also correctly classified (using neural networks). The number of images for the different speed limits is given in Table 1.

In addition to achieve a high rate of correct classification, we have made much effort at the same time to reduce the processing time. This would be highly needed in a future real-time system. The average processing time for an image is 130ms, which represents a capacity of processing about 8 images per second. This is measured on a PC with a 1.4 GHz Athlon AMD processor (512Mbyte RAM). The image filtering is the most computational demanding with about 90ms average processing time, while template matching and number recognition all together use about 40ms.

Table 1: **The number of speed limit signs classified in the sign number recognition part.**

Speed limit	Number of images
30	4
40	11
50	32
60	35
70	12
80	6

Table 2: **The results from red circle detection in video input.**

Speed limit	Number of videos	Number of videos with red circle detected
30	2	2
40	14	12
50	7	6
70	3	1
80	5	4

4.2 VIDEO

We have made some initial experiments with video input. These are so far only based on testing the detection of red circles. Table 2 shows the initial results. It is harder making detection for higher speed limits. This is reasonable since the speed of the car is faster and there is a larger risk for images being blurred. There will also be less number of images to analyse due to the higher vehicle speed. An example image from a video where it showed to be difficult to detect the red circle is given in Figure 7. The main reason seems to be that the background on this image has similar red color as the one found on the sign. Future work would consist of improving the color filtering mechanism to improve the detection. Further, number recognition should be included as well.

Figure 7: **An image from a video where it was difficult to detect the red circle.**

5 CONCLUSIONS

The paper has presented a novel approach to locate speed limit signs and detect the numbers on them. It is focused on reducing computation time at the same time as getting high recognition performance. Promising results are achieved for detecting the signs.

REFERENCES

[1] H. Akatsuka and S. Imai. Road signpost recognition system. In *Proc. of SAE vehicle highway infrastructure: safety comptatilbility*, pages 189–196, 1987.

[2] M. Lalonde and Y. Li. Road sign recognition, survey of the state of the art. In *CRIM/IIT (Centre de recherche informatique de Montreal)*, 1995.

[3] G. Wei et al. Traffic sign detection and recognition for safe driving. In Dagli et al., editors, *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems, Proc. of ANNIE'99*. ASME Press, November 1999.

[4] S.H. Hsu and C.L. Huang. Road sign detection and recognition using matching pursuit method. *Image and Vision Computing*, 19:119–129, 2001.

[5] L. Sekanina and J. Torresen. Detection of Norwegian speed limit signs. In *Proc. of the 16th European Simulation Multiconference (ESM2002)*, pages 337–340. SCS Europe, June 2002.

[6] J. Torresen, J.W. Bakke, and L. Sekanina. Efficient recognition of speed limit signs. In *Proc. of 7th IEEE International Conference on Intelligent Transportation Systems (ITSC2004)*, 2004.

[7] J. Torresen, J.W. Bakke, and L. Sekanina. Recognizing speed limit sign numbers by evolvable hardware. In *Proc. of Parallel Problem Solving from Nature VIII (PPSN VIII)*, Lecture Notes in Computer Science. Springer-Verlag, 2004.

[8] R. Thomas. Less is more [intelligent speed adaptation for road vehicles]. *IEE Review*, 49(5):40–43, May 2003.