Design and implementation of an automatic traffic sign recognition system on TI OMAP-L138

Phalguni*, Kishan Ganapathi*, Venugopala Madumbu†, Resmi Rajendran† and Sumam David*
*Department of Electronics and Communication Engineering
National Institute of Technology Karnataka, Surathkal, India
†Texas Instruments (India), Bangalore, India
Email: sumam@ieee.org

***Abstract*—This paper discusses the design and processor implementation of a system that detects and recognizes traffic signs present in an image. Morphological operators, segmentation and contour detection are used for isolating the Regions of Interest (ROIs) from the input image, while five methods – Hu moment matching, histogram based matching, Histogram of Gradients based matching, Euclidean distance based matching and template matching are used for recognizing the traffic sign in the ROI. A classification system based on the shape of the sign is adopted. The performance of the various recognition methods is evaluated by comparing the number of clock cycles used to run the algorithm on the Texas Instruments TMS320C6748 processor. The use of multiple methods for recognizing the traffic signs allows for customization based on the performance of the methods for different datasets. The experiments show that the developed system is robust and well-suited for real-time applications and achieved recognition and classification accuracies of upto 90%.**

I. Introduction

A large number of traffic accidents are caused when drivers overlook traffic signs. This fact has motivated a lot of research into the development of Driver Safety Systems. A significant component of such systems is the detection and identification of traffic signs. These systems would not only improve safety aspects of vehicles, but also help usher in an era where automated driving is possible.

Traffic signs have some unique characteristics that differentiate them from the rest of the objects in the environment. They are made of contrasting colour and are generally at a fixed height from the ground. Although these features facilitate their detection by an automated system, the recognition problem still remains a challenge due to varying lighting conditions, motion blur, fading of signs due to environmental conditions, rotation and occlusion of signs, etc. The important task here is to differentiate between identical traffic signs and eliminate processing of a large number of the objects in the background that are similar in appearance to the traffic signs to be recognized.

While a lot of research has been done in the area of Automatic Traffic Signal Recognition (ATSR) over the last two decades, very few systems have been able to satisfy the requirements of speed and accuracy at the same time. Speed and memory are conflicting requirements in real-time recognition and this is a challenge when implementing systems on a Digital Signal Processor (DSP). A lot of current systems in use are not portable onto a DSP and hence it was important to develop an algorithm that would work on a DSP. For real-time usage, the system should be capable of identifying signs that might possibly be rotated or partially occluded from a distance. These requirements motivated the development of a system that can work in noisy environments in real-time scenarios.

II. Literature Survey

A wide variety of approaches have been used to tackle the problem of Automatic Traffic Sign Recognition (ATSR). A significant number of approaches use neural networks, given their advantage of learning, which would make the system better as it is used over time. But a major problem faced when using neural networks is the difficulty they pose for implementation on a DSP in terms of time taken for training and the computational resource required. The simulation of an ATSR system using neural networks and colour segmentation is discussed by Kehtarnavaz and Ahmad in [1]. Two neural networks, one for colour segmentation and the other for classifying the invariant signatures of traffic signs, are combined to design the system. This system uses both colour and shape attributes and has some intolerance to noise.

A detailed approach for ATSR using FPGAs and multi-core System-on-Chip can be found in [2]. The solution satisfied real-time latency constraints and proves that the approach is well suited for initial analyses of designed systems. Detection times greater than one second and a significantly high false alarm were obtained when Genetic Algorithms (GA) were used [3]. The proposed algorithm performed better than other GA-based methods as well as template-matching based methods.

The fast simulated annealing algorithm, when used for the detection of the Regions of Interest (ROIs) in the image, yielded very good results in spite of the presence of noise [4]. In this approach, the object recognition problem was tackled by expressing a match between a predicted object and an image using the normalized correlation coefficient as the match metric. The templates were created on-line using transformations of the model images.

The histogram-based approach is discussed in detail in [5]. This method was implemented on the TMS320C40 DSP and tested with traffic data and also modelled occlusion. But the low accuracy (50%) of the results makes this less suitable for driver safety systems. The use of Support Vector Machines (SVMs) for ATSR is given in detail by Maldonado-Bascón et al. in [6]. The proposed recognition system was based on the generalization properties of SVMs. The system consists of three stages:

1) Colour-based segmentation;
2) Detection of signs by classification of its shape using linear SVMs;
3) Recognition of the sign using Gaussian-kernel SVMs.

The obtained results showed that the algorithm was invariant to translation, rotation, partial occlusions, etc. The use of embedded systems by Souki et al. [7] did not yield favourable results in terms of speed of computation (3.2s per frame).

Varun S. et al. [8] present an ATSR system that uses forward and reverse Euclidean distance based matching. The tree based classification gave better recognition results as signs of different shapes were under mutually exclusive categories. Hatzidimos [9] discusses the use of segmentation, edge detection and morphological analysis for the detection phase and affine transformation and cross-correlation for the recognition phase of the ATSR system. This approach yielded good results. The use of Euclidean distance based pattern matching techniques for the identification of detected traffic signs is discussed by Fleyeh and Khan in [10]. The pattern matching algorithm works with vertices of the object's shape instead of the entire image, consequently reducing the computation time. The reported accuracy is 97%.

Bo and Feng [11] describe the use of the concept of Hu moments. These moments are invariant to rotation and translation, and they obtained an accuracy of above 95%. In their work on ATSR systems, Qingsong et al. [12] used the Histograms of Oriented Gradients feature and the nearest distance method for recognition of the traffic signs present in the ROIs, which were obtained using the colour information in hue, saturation and intensity color space and the symmetry property of circles . The results obtained suggest feasibility of the approach in real-time systems.

Results reported in [9],[10],[11],[12] suggest that an optimum combination of these methods may help in implementing a good ATSR system and this forms the basis of our work.

The next section contains a brief explanation of the algorithms used in the system. Section IV describes the implementation of this algorithm on a DSP, along with a brief overview of the processor's architecture. In Section V, the results obtained are discussed.

III. System Design

Simplicity, accuracy and speed are the three factors that were considered when narrowing in on the implementation method. Neural network based methods, which require training, were avoided because of their computational complexity and training overhead. Morphological processing also requires a lot of computations. These methods were not chosen as they could not be easily implemented on a DSP. Many other algorithms involve finding feature vectors in the captured image. The efficiency of these methods is high only if the traffic sign is predominant in the frame and the image is of high resolution, which is not the case in usual road image scenarios.

TABLE I
Look up table for colour-hue range in OpenCV

Colour	Hue Range
Red	[0,10], [175, 179]
Yellow	[20,30]
Blue	[100,130]

The proposed algorithm involved simpler computations and was easy to implement on a processor. The extracted signs were saved as images of equal size in order to simplify the recognition process. Most traffic signs are red, blue or yellow in colour and hence hue is a good measure to differentiate between different colours. The input image is segmented based on colour information and the contours in the binary image are detected. The bounding box around each contour is a potential traffic sign.

The system is divided into two modules – a detection module and a recognition module. The design is based on the availability of the datasets. The first dataset [14] is comprised of on-road captured images that contains vehicles, road and traffic signs. The detection process analyses the images in the first dataset and extracts potential traffic signs. Some of the important assumptions in the detection process are:

1) The traffic signs have a red, blue or yellow coloured border.
2) The perimeter of the sign is four to forty times less than the perimeter of the input sign.
3) The maximum number of signs in a frame is 30.

Recognition was done after recognizing the shape of the signs. Fig. 1 shows the flowchart of the algorithm used for implementation.

A. Module 1: Detection

The detection process involves the extraction of ROIs from an input frame. These ROIs are further classified and recognized. The objective is to highlight all the traffic signs in the image.

1) Colour-space conversion: The RGB image was converted into the HSV colour space since HSV representation is good for segmentation based on colour information.

2) Colour-based segmentation and thresholding: The colour image was then converted into a black and white binary image in which all the pixels satisfying thresholds of the hue values given in Table I were made white and the background was made black. The pixels that are red or blue in colour are of interest as they might belong to potential traffic signs. The rest of the pixels were ignored in the next processing stage and were converted to black pixels.

3) Noise removal: Using morphological operations and denoising techniques, noise was removed from the binary images. Dilation removed small black spots and erosion removed

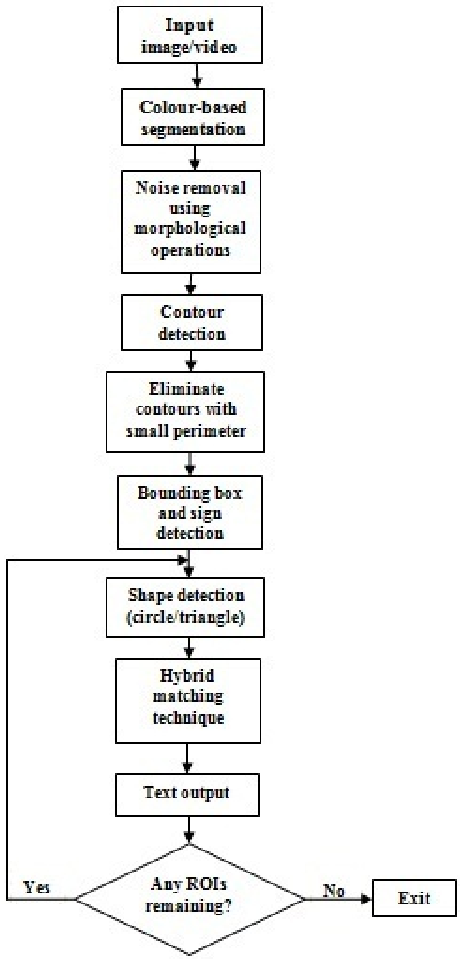

Fig. 1. Flow chart of the algorithm

white spots. The isolated pixels were removed and the main blobs were retained. This not only helped speed up the later stages in the implementation, but also helped in achieving better detection.

4) Contour detection: The binary image was then subjected to contour detection. Very small contours were rejected to reduce the chances of false detection. Smaller blobs were usually regions that were not traffic signs. These were eliminated to reduce false matches.

5) Finding the ROIs: The edge pixels in the image were assembled into contours and the detected contours in the image were considered as ROIs. The ROIs were extracted into a new image and used as the input for the matching algorithm. The bounding box of minimum area enclosing the contour was considered as the ROI.

6) ROI Validation: Aspect ratios of the detected contours were considered in validating the sign. The ROI was considered valid only if the aspect ratio was between 0.9 and 1.2 because the ROIs are roughly square in shape. Other ROIs were considered invalid and removed from further processing. The validated ROIs were resized to 64x64 resolution before storing them in memory.

False sign detection was minimized using an optimum choice of colour thresholds, maximum and minimum contour perimeter, aspect ratio, number of iterations of noise removal by morphological opening, etc. This is explained in the next section.

B. Module 2: Recognition

The recognition stage is aimed at identifying the shape of the sign in the 64x64 image and recognizing the sign. Based on the Hough transform, the shape of the traffic sign can be detected. Hough transform for circles classifies the signs into circles and non-circles. Hough transform for lines can be used to distinguish further between triangles and octagons. If the number of lines detected is 8 or more, the shape is considered as an octagon. If the number of lines is less than 3, it is classified into the non-sign category. Even if the ROI was classified as circular, triangular or octagonal, an image belonging to the non-sign category was included in the database so that any resemblance to that particular image can be used to classify the input image into the non-sign category. The red tail-lights of vehicles on the roads was a common source of such false positives.

1) Hu moments: The moments of an image are the statistical expectation of certain power functions of its pixel intensities. The central moments refer to moments calculated about the images's centroid. The Hu invariant moments h_i $(i = 1, 2, ...7)$ are linear combinations of the central moments. The idea here is that by combining the different normalized central moments, it is possible to create invariant functions representing different aspects of the image in a way that is invariant to scale, rotation and reflection. If h_i^A and h_i^A are the Hu moments corresponding to image A and B, *I(A,B)* is obtained by:

$$I(A, B) = \sum_{i=1}^{7} \left| h_i^A - h_i^B \right| \ . \tag{1}$$

The comparison method tries to match shapes by finding the minimum value of *I(A,B)*, the distance between two images.

2) HSV histogram: The technique involves finding the similarity between histograms of the captured sign in HSV space for hue and saturation values against the database sign. In our system, the Correlation method was used to determine similarity.

3) Histogram of Gradients: The Histogram of Gradients (HoG) method [12] uses the shape information to find similarities between the ROI and the database image. The derivative at a particular pixel has two components: the magnitude and the orientation. The potential region of the image was converted to a grayscale image, followed by equalizing its histogram. The image was resized to 64x64 resolution. The image was then divided into sub-images of size 16x16 and the gradient was computed using Sobel kernels of size 3x3 for each sub-image.

The angle at a particular pixel is in the range -90 to +90 degrees. This was adjusted to fall in the range of 0 to 360 degrees. This range was divided into 8 bins and the magnitude

Fig. 2. Composite image made of individual triangular signs

of the derivative was accumulated in the bins according to the value of the angle at a particular point. All the units were linked to form a feature vector of 128 (4x4x8) dimension. The feature vector was normalized to remove the influence of illumination and edge contrast. The similarity between the feature vector obtained by the sum of the squares of the differences or dot product method gives an estimate of the best match among the database images.

4) L2 distance: This method finds the Euclidean distance between the ROI and templates. In case of a good match, the L2 distance would be low.

5) Template matching: In this method the ROI was made to slide through a reference composite image, made up of multiple images of traffic signs as in Fig. 2, and a comparison of the overlapped patches of size w x h (of the composite image) against the ROI was done.

The origin of the best match might be located anywhere in the composite image. A correction factor was added to the co-ordinates of the origin so that the nearest sign was considered as the best match. All the individual images in the composite image were of size 64x64. The x and y co-ordinates were corrected to the nearest multiple of 64.

The feature vectors obtained from the Hu moments, HSV histogram, template matching, L2 distance and HoG methods were combined to get a weighted average score for each sign in the database. The weight assigned was proportional to the relative accuracy of each method in isolated testing. The sign in the template database with the maximum number of weighted votes was considered as the sign in the ROI.

IV. System Implementation

A. Datasets

The first dataset used in detection stage contained 500 images of 1280x960 resolution, captured on road with traffic signs occupying less than 20% of the frame area [14]. Images from this dataset were the inputs to the system. Images from the second dataset [15], of size 64x64, were used as reference images in the recognition stage. The recognition module used 24 images, with the traffic sign occupying more than 80% of each image's area.

B. OpenCV Implementation

Intel's OpenCV is coded in optimized C and can take advantage of multi-core processors [13]. The traffic sign recognition algorithm was coded in OpenCV for optimizing and porting the algorithm onto the DSP.

The control parameters help in optimising the code and improving the accuracy of the system. The values of the parameters depend on the dataset being used. The similarities between the signs in the images from the first and second datasets influence the range of values that were optimum for use in the implementation, and minor modifications were made based on results obtained during testing.

1) Hue thresholds: These are the range of hue values that need to be considered in segmentation of the traffic signs. Red falls in the range (0 to 10) and (175 to 179), and blue in the range of (100 to 130). Slight changes in the range of these values led to an increase in false positives or false negatives.

2) Number of iterations in the dilation and erosion processes: This depends on the amount of noise in the input image frame, which in turn depends on the environment in which the image has been captured. It was found that one iteration was sufficient for most cases.

3) Number of contours: Assuming that the number of potential signs cannot be infinite, this number was set to 30 to limit the computational load.

4) Minimum perimeter of a contour: The traffic sign occupied an area that was between 1/40 and 1/4 of the perimeter of the frame in the dataset considered. This parameter may be varied according to the dataset being used.

5) Number of database signs: Eight circular and twelve triangular signs were present in the database. The composite image in template matching was composed of a 2x4 array of circular signs and a 3x4 array of triangular signs.

6) Number of lines: For non-circular signs, number of lines in the image differentiates between triangles and octagons. For the dataset under consideration, the sign was considered as octagon if the number of lines was greater than 7.

C. DSP Implementation

The OMAP-L138 Experimenter Kit, with a low-power dual core 300MHz C6748 DSP and an ARM926EJ-S 32-bit RISC MPU was used in the DSP implementation. Only the C6748 side of the device was used. The XDS100v1 embedded emulation was chosen which made use of the C6748 DSP part of the kit.

On the DSP, the detection module required about 40 milliseconds and the recognition module required 250 milliseconds to run completely. Hence the maximum total time in real time implementation can be estimated to be about 300 milliseconds. Therefore the output frame rate was 3fps. The entire process is input-dependent. The time taken depends on the number of pixels that are red or blue in colour, number of contours detected, value of pixel at each point in the grayscale image, etc. The maximum implementation time reported is under the assumption that an image frame can have a maximum of 100 ROIs. Detection and recognition was carried out for 500 images and the minimum, average and maximum execution times were noted. If a vehicle moves at the speed of 80 km/h, it covers a distance of 22 metres per second. If the detection is

Fig. 3. Input image and image after thresholding

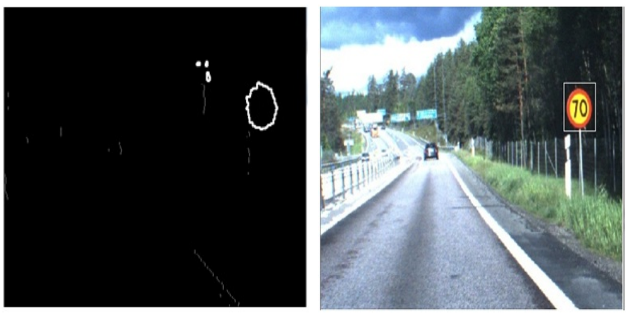

Fig. 4. Results after contour detection and finding the ROI

carried out at rate of 3fps on DSP, the distance travelled before the next frame is captured is about 8 metres. This is sufficient for real time implementation, and tracking is not required as there is negligible correlation between two successive frames. If the execution time increases, there are chances of missing the traffic signs which are within a distance of 8 metres from the previous captured point.

V. Results

The detection phase extracted 90% of the signs in the input images. The first database with 500 images [14] was used for detection. The time required by the DSP at 300MHz for executing the detection module is tabulated in Table II. The average execution time for a traffic sign is given in Table III. To test the efficiency of the recognition module 500 test images were used, with about 15 images per sign. Table IV shows the rate of accuracy for the various matching techniques. For each technique, a range has been mentioned as the recognition rate is different for different signs. The overall rate can be considered as the average in the range. Table V contains the profiling results for signs of different shapes.

Fig. 5. Real-time image with detected signs, result of template matching

Fig. 6. Result of template matching for stop sign

TABLE II
Profiling results for the detection module on OMAP-L138

Data parameters		Execution time (ms)
Image size	No. of ROI	
128x128	1	0.57
128x128	2	1.08
480x360	1	3.92
480x360	2	9.13

The results of the detection phase are shown in Figs. 4-6. The recognition result for template matching is shown in Fig. 6.

Many ROIs do not have a traffic sign. The detection module was designed to have a large number of ROIs so that potential signs were not missed. ROIs that do not contain traffic signs were classified as non-circles and further into the non-sign category. Though the false positives rate was 20%, the effective rate after classification was reduced to 5% due to the correct classification of ROIs that were not signs. The chances of missing a sign in the detection stage was only 3%.

VI. Further Work

There are certain techniques that could improve the accuracy and speed, decrease the memory requirements and make the ATSR system more robust for real time applications.

To increase speed and performance, both the cores of the OMAP-L138 – DSP and ARM could be used. External emulation provides various options like DSP only, ARM only or DSP + ARM. This technique would achieve a significant reduction in processing time. With more frames processed per second, the chances of missing a sign can be reduced.

Enabling the cache of the C6748 to access frequently required data like the input image and database signs is another method to speed up execution.

TABLE III
Average execution time for recognition on OMAP-L138

Method(s)	Execution time (ms)
Hu moment method / Template matching / L2	100
HSV histogram	120
Histogram of Gradients	134
Hu moment, Template matching, L2	130
Hu moment, Template matching, L2, HoG	250

TABLE IV
RATE OF ACCURACY FOR VARIOUS RECOGNITION METHODS

Method	Lower rate	Higher rate	Overall rate
Hu moment based matching [11]	20%	40%	30%
HSV histogram based matching [5]	30%	65%	47%
Histogram of Gradients approach [12]	40%	70%	55%
L2 distance based technique [10]	35%	70%	52%
Template matching [9]	60%	90%	75%

TABLE V
PROFILING RESULTS (EXECUTION TIME) FOR RECOGNITION OF SIGNS OF DIFFERENT SHAPES ON OMAP-L138

Method(s)	Execution time (ms)		
	Circle - No overtaking	Octagon - Stop	Triangle - Left bend
Hu moment method	119.79	116.05	112.74
Template matching	114.13	123.39	113.73
L2 (Euclidean distance)	73.66	103.37	112.96
HSV histogram	121.78	125.15	116.04
Histogram of Gradients (HoG)	146.21	153.04	157.47
Hu moment, Template matching, L2	128.42	128.66	130.04
Hu moment, Template matching, L2, HoG	250.88	259.71	212.41

Other approaches that involve learning could also be used, provided the training is performed on some other machine and only the final data is given as input to the DSP.

In our system, we do not track the traffic signs as there is negligible correlation between two successive frames when considering the processing time for each frame. Kalman filters and other methods could be used to track and validate the traffic signs. This becomes a necessity when the processing time decreases. As the system becomes faster, the correlation between successive frames will increase and using tracking would help reduce the number of ROIs that need to be recognized.

VII. CONCLUSION

We have developed a practical ATSR system using an efficient algorithm coded in OpenCV and implemented it on a DSP. The various recognition methods have been compared based on the number of CPU clock cycles required for execution.

The proposed system does not require any training, so it can be optimized and modified easily as per the dataset in hand and changing conditions. The real time constraints are met and the system works as a stand-alone application on the DSP without posing any problems in terms of speed and memory requirements.

ACKNOWLEDGMENT

The authors would like to thank Texas Instruments Ltd. (Bangalore) and Mr. K. Aravindan for their support to this work, which carried out as an undergraduate project. We would also like to thank the reviewers, whose suggestions have improved the quality of the paper.

REFERENCES

[1] N. Kehtarnavaz and A. Ahmad, “Traffic Sign Recognition in Noisy Outdoor Scenes”, IEEE Intelligent Vehicles Symposium (IV), Detroit, MI, USA, Sept. 1995, pp. 460-465.

[2] M. Müller, A. Braun, J. Gerlach, W. Rosenstiel, D. Nienhüser, J. Marius Zöllner and O. Bringmann, “Design of an Automotive Traffic Sign Recognition System Targeting a Multi-Core SoC Implementation”, EDAA 2010, pp. 532-537.

[3] A. Soetedjo and K. Yamada, “An Efficient Algorithm for Traffic Sign Detection”, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 10, No. 3, 2006, pp. 409-418.

[4] M. Betke and N.C. Makris, “Fast Object Recognition in Noisy Images Using Simulated Annealing”, Proceedings of the Fifth International Conference on Computer Vision (ICCV), June 1995, pp. 523-530.

[5] L. Estevez and N. Kehtarnavaz, “A real-time histographic approach to road sign recognition”, Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation, San Antonio, Texas, 1996, pp. 95-100.

[6] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno and F. López-Ferreras, “Road-Sign Detection and Recognition Based on Support Vector Machines”, IEEE Transactions on Intelligent Transportation Systems, Vol. 8, No. 2, June 2007, pp. 264-278.

[7] M. A. Souki, L. Boussaidand M. Abid, “An embedded system for real-time traffic sign recognizing”, 3rd International Design and Test Workshop, 2008, pp. 273-276.

[8] Varun S., S. Singh, S. Kunte R., S.R.D. Sudhaker and B. Philip, “A Road Traffic Signal Recognition System based on Template matching employing Tree classifier”, International Conference on Computational Intelligence and Multimedia Applications 2007, pp. 360-365.

[9] J. Hatzidimos, “Automatic Traffic Sign Recognition in Digital Images”, Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - ICTAMI 2004, Thessaloniki, Greece, pp. 174-184.

[10] H. Fleyeh and T. Khan, “Pattern Matching Approach Towards Real-time Traffic Sign Recognition”, 2010 International Conference on Multimedia Computing and Information Technology (MCIT), Sharjah, pp. 57-60.

[11] Z. Bo and Z.T. Feng, “Research on the Location and Recognition Methods for Traffic Signs”, Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications, 2009, pp. 425-428.

[12] X. Qingsong, S. Juan and L. Tiantian, “A Detection and Recognition Method for Prohibition Traffic Signs”, International Conference on Image Analysis and Signal Processing (IASP), 2010, pp. 583-586.

[13] Gary Bradski and Adrian Kaehler, Learning OpenCV, OReilly Media, 2008.

[14] Traffic Signs Dataset, Computer Vision Laboratory, Department of Electrical Engineering, Linköping University – http://www.cvl.isy.liu.se/research/traffic-signs-dataset

[15] The German Traffic Sign Recognition Benchmark, IEEE International Joint Conference on Artificial Neural Networks (2011), Institut für Neuroinformatik – http://benchmark.ini.rub.de/index.php?section=gtsrb&subsection=dataset