

Speed Limit Road Sign Detection and Recognition System

Fakhradeen Hamid Ali
Engineering College/ Mosul University
Mosul, Iraq

Mohammad Haqqi Ismail
Engineering College/ Mosul University
Mosul, Iraq

ABSTRACT

Traffic signs in general and speed limit signs in particular are considered one of the most important means of traffic safety, and the aim of the current research is to design a system that detects and recognizes speed limit sign with high accuracy and high processing speed. At the beginning, red color objects are detected from the image and after finding the red color signs the circle is determined using Hough's transform then from inside the circle, the numeric part from the circle image is extracted. Digital circle images are segmented to extract the number alone, and then these numbers are recognized by a trained neural network. Neural network achieved a success rate in recognition reached to 98.9%. Parallel programming concept is used to reduce the execution time using OpenMP and OpenCL programming. The study showed that the total execution speed according to the designed scheme to run the speed limit sign detection and recognition by using a mix of central processing unit with multi cores and graphics processing unit is 65 frames/sec for complete images and 90 frames/sec when cropping the effective part from the total size of the image. Recognition system is capable of recognizing the sign even if the vehicle speed exceeds 120 km/h.

Keywords

Road sign detection, neural network, number recognition, color space, OpenMP, OpenCL, multicore, graphics processors, color Segmentation.

1. INTRODUCTION

Vehicles are informed with the maximum speed limit which is allowed in a certain road through automatic finding of the speed limit sign using image processing where the speed limit sign is detected in a red ring which surrounds the sign in a circle form, where, first the image is segmented in color to extract the red part and removing all noise and unnecessary small objects using morphological opening operations, smoothing and edge detection. Circular Hough's transform is adopted to detect the circle, which represents our target, then cropping it off out of the image. After that recognition operation starts which includes extracting the black color number then making processing operations including truncation, normalization and smoothing the number to apply number recognition using neural network [1,2,3,4]. The system accuracy is not only detecting and recognizing the speed limit sign correctly but it considers the execution speed of detection and recognition which is important in real time applications. Modern architectural in device machine to pass executive time problem

load is used to improve the system performance through multi cores central processing units and graphics processing unit and the possibility to perform tasks in parallel with each other

[5,6,7,8]. The aim of the current research is to design and implement speed limit sign automatic detection and recognition system with high accuracy and processing speed, as well as, checking and validating the system performance.

2. SYSTEM DESIGN

Figure (1) shows a general flow chart of the speed limit sign detection and recognition system which includes image or frame reading, image color segmentation, speed limit sign detection, recognition and result output.

2.1 Image or Frame Reading

Logitech C920 web-camera is used, where it was fixed inside the front glass of the vehicle to monitor the outside environment with a resolution 640 x 480 pixel and a speed 30 frames/second.

2.2 Image Color Segmentation

It is the operation of recognizing the limit objects in the image from its background where three color spaces were used, they are RGB, HSV and YCrCb, and based on the color information, it is possible to separate the speed limit signs from the rest of the image parts. This is accomplished by generating a binary image includes the speed limit sign [9]. Logical AND operation was used to implement image segmentation algorithm taking three color space values; red, green and blue in RGB space and hue, saturation, and value in HSV space, and values of each of Y, Cr, and Cb in YCrCb space. Color segmentation algorithm was implemented by transferring the image from RGB into HSV using standard equations [10]. To extract red color from HSV, multi threshold value was used, where red color can be detected in all values of high and low hue. The extraction of red color from HSV produces better results than using RGB and YCrCb because images are able to be segmented by using same thresholds and this is impossible with RGB and YCrCb, And if a global threshold for all different lighting conditions was used, the outcome image from the red color segmentation with RGB and YCrCb would be not good and it contains undesired objects. Table (1) shows the thresholds used in different applications to extract red color.

2.3 Speed Limit Sign Detection

2.3.1 Removing Small Objects: noise and some unwanted pixels are removed from image using morphological opening operation and this will lead to reduce the processing time in the next stages [11].

2.3.2 Smoothing

To get rid of the negative effect of some noise which is done by using one of smoothing filters which is called gauss filter [12].

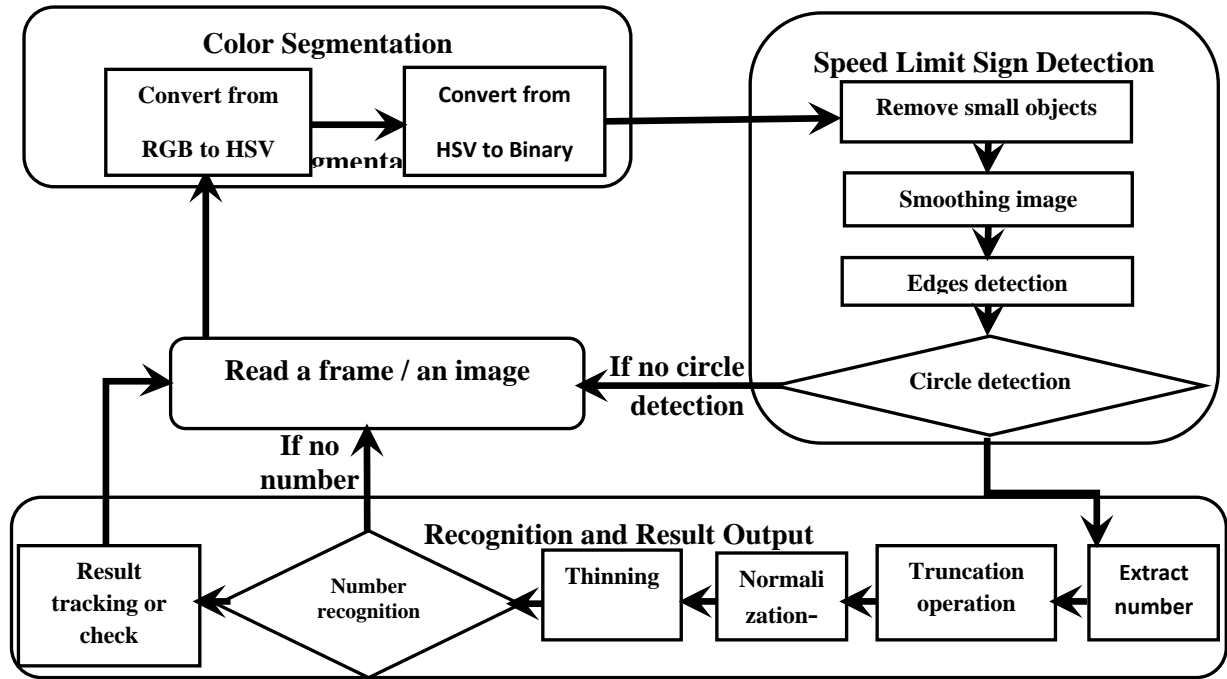


Fig (1): General flow chart of speed limit sign detection and recognition system

Table (1): Thresholds used to extract red color.

Color space	Sunny day	Rainy day	Fog	Evening	Night
RGB	178<R<255 0<G<130 50<B<100	60<R<255 0<G<66 0<B<55	60<R<255 0<G<65 0<B<55	45<R<68 0<G<50 0<B<45	150<R<255 50<G<150 50<B<150
YCrCb	30<Y<255 168<Cr<220 70<Cb<110	30<Y<255 135<Cr<168 117<Cb<130	30<Y<255 134<Cr<145 110<Cb<130	30<Y<255 134<Cr<147 120<Cb<130	30<Y<255 165<Cr<200 88<Cb<118
HSV	0<H<10 60<S<255 60<V<255	0<H<10 60<S<255 60<V<255	0<H<10 60<S<255 60<V<255	235<H<255 60<S<255 60<V<255	0<H<10 60<S<255 60<V<255

2.3.3 Edge Detection

where many extra information in the image are removed while keeping edges that determine speed limit sign board by using Canny method.

2.3.4 Circular Sign Detection

In this study suitable radius range of image speed limit sign was identified, where it needs a number of tests to monitor speed limit sign by using recorded video by the camera fixed inside the vehicle, where 6 pixels are identified as the minimum radius and 30 pixels as the maximum radius of the circular image sign. Determining radius is made after detecting the circle and fixing its center. Hough transform 2-IHT[13] was used in which the center of a circle must lie on the normal of each point on circumference of a circle, the common intersection point of these normals is actually the center of the circle [14]. The normal can be drawn on any

edge point in two sides with 180 angle for not knowing the direction of circle center [13]. The length of each normal varies from minimum to maximum radius which can be determined. Two-dimension array is required to accumulate votes along the normal for each edge point. To determine the circle radius, the distance from each edge point to the candidate center is calculated then a histogram is

produced for these distances and depending on the greater iteration to represent the wanted circle radius [13]. Selecting the suitable value of the accumulator array threshold is necessary to avoid the selecting of low values which leads to detect more circles and the selecting of high values which are not enough to detect the circle. Figure (2) shows results of image color segmentation and speed limit sign detection.

2.4 Recognition Stage and Output Results

The purpose of this stage is to extract the image of speed limit sign from the original image then several processing steps were done to obtain the black number inside the sign. Many processing stages, including truncation, normalizing and smoothing to prepare the number in a suitable form for recognition stage by neural network. Using the suggested method to extract the black color inside the sign, the sign was extracted from the original RGB image and then it was converted into gray, then threshold value was used to convert it into binary image. The threshold value is a mean of pixels of the region that contains the speed limit number.

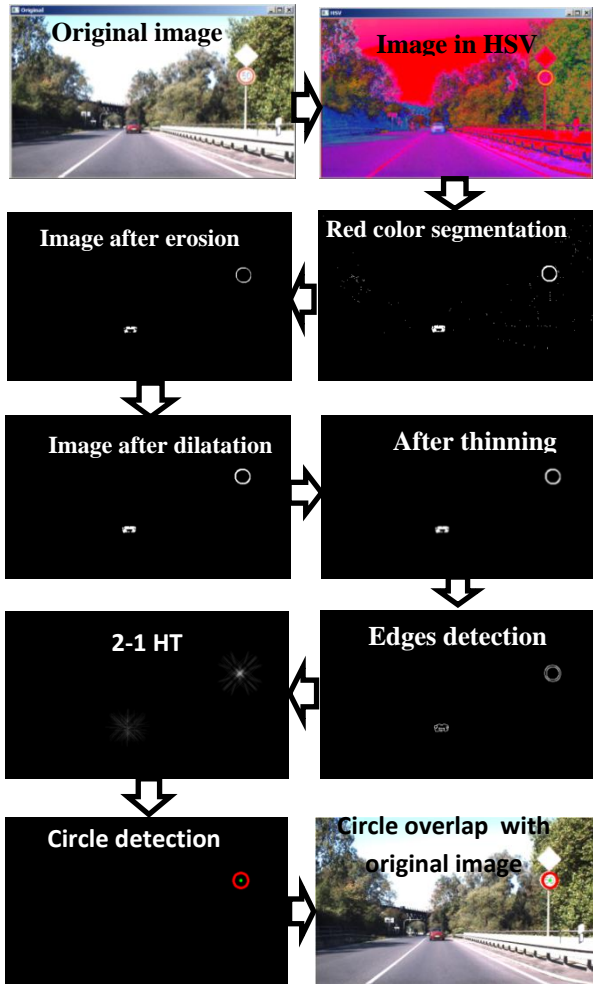


Fig (2): Results of image color segmentation and speed limit sign detection.

Figure (3) shows the steps of extracting the black number inside the sign by the proposed method. A mask was made depending on the information given by the circle detection represented with its radius, where a black square image was made with a side length equal to the circle diameter and the mask includes a central white circle whose radius is a little lower than the circle radius which is represented as H in figure (3) and it was named the white mask. Its purpose is to help in processing the dark signs in the bright images in addition to calculating the mean of pixels. Also another mask was made depending on the same above mentioned information, where a white square image was made with a side length the same as the circle diameter and the mask includes a central black circle and its radius is a little lower than the circle radius presented as I in figure (3) and it was named as a black mask. Logical AND operation was applied between the white mask and the gray image, where the

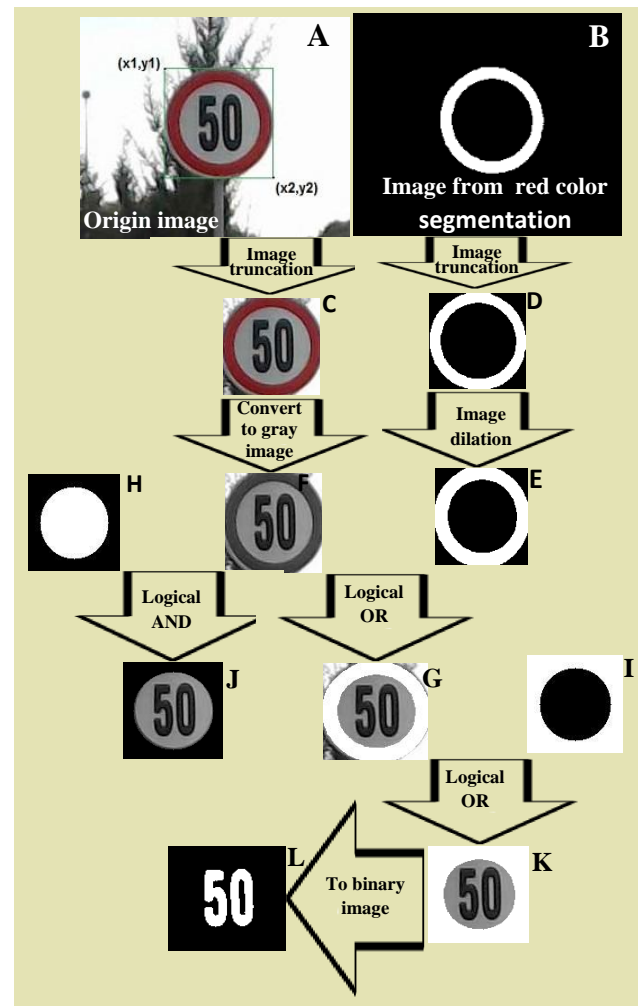


Fig (3): The depended steps in extracting the black number inside the sign by suggested method.

produced image by this operation has been used in calculating the mean, and the purpose of this operation was to make the image pixels surrounded the speed limit number equals zero so as not to affect the image pixels mean calculation. Figure (4) shows some different examples by using the proposed method to extract the black color inside the sign under different lighting conditions. Several methods were tested to extract the black image such as the different color spaces RGB, HSV and YCrCb in different lighting conditions but it was fruitless.

2.4.1 Truncation Image Operation: its purpose is to determine beginnings and ends of the number by using horizontal and vertical projections for the segmented number. The horizontal projection is done by collecting white pixels of each column in one dimensional array, and the vertical projection is done by collecting white pixels of each row in one dimensional array. These projections tell the image content of the white pixels along two axes.

2.4.2 Normalization: This operation is used because speed limit sign images sizes are different and this is due to their locations and their distances from the camera fixed inside the vehicle. So there is an urgent requirement to adopt certain image dimensions because the size of the inserted number is possible to be greater or even smaller than the expected, so normalization is made to qualify the number for the next stage. The adopted normalization size (40 x 40) is

used before input to the neural network.

2.4.3 Thinning: this operation is needed for thinning the number image produced by normalization continuously till obtaining the skeleton of the wanted number image [15]. This operation has a positive effect on the accuracy of recognition depending on a simple and easy form for each number so that it leads to fast recognition. Figure (5) shows the number image after truncation, normalization and thinning operations.



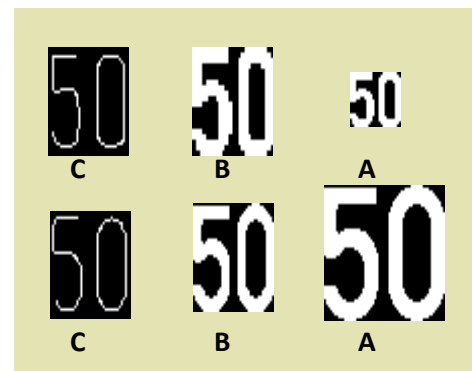
Fig (4): Different examples using the proposed method to extract the black color inside the sign under different lighting conditions.

2.4.4 Artificial Neural Networks

This method [9,16] was used to recognize speed limit signs numbers where a data base includes 2909 images of speed limit sign has been adopted to train the network and testing its achievement. The neural network was designed, trained and tested to recognize 8 types of speed limit (20, 30, 50, 60, 70, 80, 100, and 120). The back propagation algorithm has been used. Neural network design includes designs for input, hidden and output layers in addition to selecting the activation function. According to the input image size, 1600 input cells were used and the array of the speed limit number image was converted from 40*40 image units to a vector contains 1600 pixels. Hidden layers number and cells number in each layer are determined by experiment, where number of layers and number of cells in each layer determines the network ability to training and recognition. The number of hidden layers and the number of cells in each layer is varied until obtaining the best accuracy in recognition, in addition to training speed. Through many attempts, the best performance of the network was obtained using one hidden layer with 30 cells. And the output layer according to the cases that need recognition has 8 cells and they are represented with speed limit signs. Neural network achieved a success rate in recognition reached to 98.9%.

The input vector for the network is the binary image produced from the thinning operation. The images for training the

network have been selected to contain speed limit sign images with different sizes tilting with vertical by different angles. To reduce the probable mistakes produced from speed limit signs recognition, they are checked before warning the driver. Single successful recognition of one video frame does not confirm the absolute recognition of the speed limit sign, and speed recognition system continues in tracking these signs in next frames and this leads to increase the accumulated probability when reaching to a certain threshold where speed is showed to warn the driver. In this study a threshold value with three repetitions having the same recognition result for sequence frames was decided to recognize the speed limit sign absolutely. This selection has been checked by testing a video record includes 19 speed limit signs in addition to 16 other road signs, and the result showed that when doing repetition for three sequence times, total system accuracy reaches 100%. This operation reduces the false results and confirms the right recognition, it adds solidity to the designed recognition system.



**A. After Number Truncation.
B. After Number Normalization
C. After Number Thinning**

Fig (5): Number image example after truncation, normalization and thinning.

3. EXECUTION TIME

Execution time of all speed limit sign detection and recognition stages was measured with different processing methods using multi cores CPU and GPU to process data in parallel where in parallel programming Open Multi-Processing (OpenMP) used with CPU and Open Computing Language (OpenCL) used with CPU or GPU. Two laptops were used to implement all algorithms of the speed limit sign detection and recognition system, tables (2) and (3) show device specifications of each used CPU and GPU for the two used machines.

3.1 Execution Time of Image Color Segmentation

Table (4) shows time measurement results expressed by milliseconds (ms) of color segmentation operations,

Table (2): devices specifications of CPU for the two used Machines.

CPU	Model	Machine one	Machine Two
		Intel(R) Core(TM) i7-2670QM	Intel(R) Core(TM) i5-3210M
Architecture	Sandy Bridge	Ivy Bridge	
Number of physical cores	4	2	
Number of logical cores	8	4	
Clock speed	2.20 GHz	2.50 GHz	
Operating system	Windows7 x64 bits	Windows7 x32 bits	

Table (3): devices specifications of GPU for the two used Machines.

Property	Machine one	Machine Two	
	AMD Radeon HD 6770M	Intel(R) HD Graphic 4000	NVIDIA GeForce GT 640M
Architecture	Terascale 2	Ivy Bridge	Kepler
processor cores	480	16	384
MAX WORK ITEM SIZE	256 x 256 x 256	512x 512 x512	1024 x 1024 x 1024
MAX WORK GROUP SIZE	256	512	1024

Table (4): Execution time (ms) of color segmentation operations

Color Segmentation	Machine one						Machine Two		
	OpenCL		LUT by OpenMP	LUT CPU	OpenMP	Sequential	OpenCL		
	GPU AMD	CPU					HD Graphic	GPU NVIDIA	CPU
RGB To HSV	2.38	1.61	0.33	1.10	3.14	10.87	6.02	1.00	2.51
HSV To binary	1.20	0.38	0.20	0.58	0.25	0.774	4.39	0.60	0.81
RGB to HSV then to binary	1.39	1.43	0.36	1.79	2.44	9.62	4.76	0.79	2.17

converting RGB to HSV, converting HSV to binary image, and converting RGB to HSV then to binary image with different processing methods including CPU sequential programming,

OpenMP parallel programming, Lookup Table (LUT), LUT with OpenMP, parallel programming OpenCL (GPU AMD and CPU) by using Machine one then parallel programming OpenCL (HD Graphic and GPU NVIDIA and CPU) by using Machine two. Generally the table shows that the lowest execution time when processing with LUT using parallel programming OpenMP and that comparison between the executive time of image conversion from RGB into HSV and image conversion from RGB to HSV then to binary image, in the second case is lower than first case except LUT and this is due to that in the second case RGB is converted from RGB with its three components into binary image with one component. While with LUT, converting from RGB into HSV, this data is read by memory address values R,G and B, and that converting from RGB into binary image includes converting from RGB into HSV in addition to converting into binary image. To compare between parallel programming performance of CPU of each OpenMP and OpenCL and sequential programming by using machine one, where table shows that speedup rate in OpenCL is about 3.9 while OpenCl is 6.7 when converting color image RGB to HSV then to binary image.

3.2 Execution Time of Speed Limit Sign Detection

Table (5) shows time measurement results expressed by milliseconds of sequence stages in detecting speed

limit signs which include erosion, dilation, smoothing, edges

detection and Hough transform by different processing methods

including CPU sequential programming, parallel programming

OpenMP, parallel programming OpenCL (GPU AMD and CPU) by using machine one then parallel programming OpenCL (HD Graphic and GPU NVIDIA and CPU) by using machine two. The table in comparison with HD Graphic, GPU NVIDIA, GPU AMD shows that the fastest execution time is with screen card GPU NVIDIA then in GPU AMD then HD Graphic and one of the reasons behind this is that each work group is implemented in compute unit at the same time and according to the input image size 640 x 480, and when choosing the maximum work group size represented with the dimensions local size, where in NIVIDIA the number of work group is 300 and in AMD the number of work group is 1200 and in HD Graphic the number of work group is 600 and according to the available compute units number in each architecture the repeated processing operations of the image is 150 in NIVIDA because it contains 2 compute units[17], the repeated processing operation of the image is 200 in

AMD because it contains 6 compute units [18], and the repeated processing operation of the image is 38 in HD Graphic because it contains 16 compute units [19]. Also processing elements number in each compute unit is 192 in NIVIDA, 80 in AMD and 1 in HD Graphic. When compute units in GPU are more, it doesn't necessarily means that processing is faster, where number of compute units is changed according to the different machine architectonics and their types [19].

3.3 Executive Time of Recognition and Output Results

Table (6) shows execution time measurement results expressed by milliseconds of recognition and output stages which include extraction, truncation, normalization, thinning and number

recognition (convert image from 2D to 1D and neural network) by processing methods including CPU sequential programming, and parallel programming OpenMP for the two used Machines. To compare the two Machines performance, the table shows that the execution time of sequential programming in machine two is less than in machine one because the CPU Clock speed of machine two is greater. The executive time of parallel programming OpenMP in machine one is less than in machine two because the number of cores in machine one are eight while

they are only four in machine two. To compare between sequential programming and parallel programming using OpenMP, the table shows that execution time of sequential programming is greater than OpenMP in all stages of recognition and output except extraction operation and vice versa, where extracting the number includes many processing stages and needs memory synchronization which consumes time in addition to some overhead which it results from start up the threads and assigning work to them. Also in the process of calculating the pixel mean which used a variable of reduction type in OpenMP programming where it should accumulates from All-threading inside the parallelism loop, and this leads to the consumption of time which it considered a large proportion to the amount of load to be processed.

Table (5): Execution time (ms) of sequence stages in detecting speed limit signs

Speed Limit Sign Detection	Machine one				Machine Two		
	OpenCL		OpenMP	Sequential	OpenCL		
	GPU AMD	CPU			HD Graphic	GPU NVIDIA	CPU
Erosion	1.44	1.48	0.22	0.80	10.00	0.72	2.27
Dilate	1.48	1.80	0.35	1.01	10.01	0.71	2.32
Opening	2.35	3.22	0.58	1.84	19.61	1.20	4.18
Smoothing	1.53	3.55		3.26	7.75	1.16	4.67
Edges detection	4.39	12.5		11.90	14.71	4.10	17.54
Circles detection	11.77	17.9		25.00	21.28	7.81	28.57

Table (6): Execution time (ms) of recognition and output stages

Recognition and Result Output		Machine one		Machine Two	
		OpenMP	Sequential	OpenMP	Sequential
Extract Number		0.06	0.04	0.06	0.04
Truncation Operation		0.03	0.09	0.05	0.09
Normalization		0.02	0.06	0.03	0.06
Thinning		0.47	0.75	0.56	0.69
Number Recognition	Convert image to vector	0.01	0.03	0.01	0.03
	NN		0.10		0.10

3.4 Total Execution Time

Figure (6) shows the designing flowchart of operating speed limit sign recognition system with its two parts: host and device depending on the execution time of different stages in speed limit sign

detection and recognition system the flowchart of the system

which is designed through many possible implemented attempts where it includes implementing color segmentation by using LUT with OpenMP , also implementing the stage of speed limit sign detection with its all stages: erosion, dilation, smoothing, edges detection and circle detection by using GPU and implementing stages of sign recognition and output results where extracting number was done by sequential programming while truncation, normalization, and thinning number by OpenMP followed with neural network.

Table (7) shows the total execution time of speed limit sign

recognition system expressed by frames/sec with different devices for the two machines. The system input is a sequence of videos frames size 640 x 480. Speed limit traffic sign is recognized by system with distance 21 meters, also is detected by system with distance 35 meters, relying on absolute recognition with three repetitions with the same recognition result of the sequence frames, and this leads to recognize a video frame at least during 7 meters, where the maximum allowed speed on the highway roads 120km/hr which means 33.33m/sec. Therefore the required frames mean for this system is 4.76frames/sec. and according to the shown executive speeds in table (7) and for different devices and the required frames mean, recognition system is able to recognize the sign even with speed greater than 120km/hr.

Table (7): The total executive time expressed by frames/sec with different devices

Machine	Device	Host	Total executive speed in using effective image part	Total executive speed
Machine	AMD	CPU	90	65

one	CPU	CPU	38	22
Machine Two	NVIDIA	CPU	60	50
	HD Graphic	CPU	23	13
	CPU	CPU	18	13

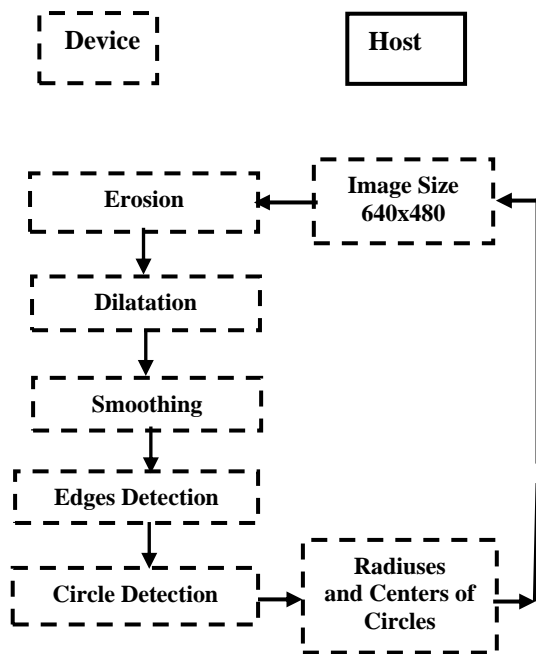
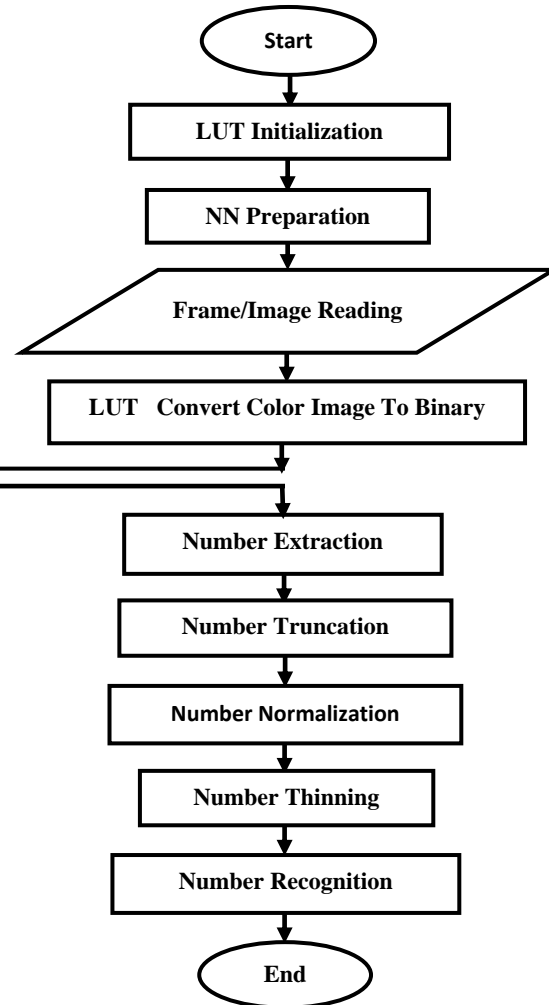


Fig (6): Flowchart of operating speed limit sign recognition system.



4. CROPPING THE EFFECTIVE PART OF IMAGE

To increase performance speed during processing operation, the effective part of the image which contains the speed limit sign with high probability is cropped and used in order to extract the sign which represents half of the middle image size which was taken by camera, where 20% of the top image size is excluded and 30% of the bottom image too. Figure (7) shows image representing a video frame containing the sign in the effective

part of the image after cropping 50% of the image size. This idea was applied on the recorded and downloaded videos by Google search and there was a possibility that the image will be in the middle part of the image reaches 100%. Execution speed when cropping half of the image was measured and according to the design plan of operating speed limits sign detection and recognition system with its two parts: host and device. The table

(7) shows that there is an increase in the total execution speed

due to cropping half of the image varies from 20% to 77%.

5. CONCLUSION

In this paper we have presented a real-time speed limit sign automatic detection and recognition system with high accuracy. HSV produced better results in the extraction of red color from road sign for all different lighting conditions. A method was suggested to extract the black color inside the sign and this method is capable of extracting the number with different environmental conditions. Neural network was trained to recognize the speed limit traffic sign with a success rate reaches to 98.9%. The lowest execution time when processing of image conversion from RGB into HSV is by using LUT. When comparing between graphical processing units AMD, NVIDIA and HD Graphic, the smallest execution time is with NVIDIA

then AMD then HD Graphic for the speed limit sign detection (erosion, dilation, smoothing, edges detection and circle detection) although NVIDIA has less compute units. According to the total execution time of speed limit sign detection and recognition system using different devices

depending on sign distance of recognition, the system is able to recognize the sign even with a vehicle speed greater than 120km/hr. The total execution time and according to the designed plan to operate the speed limit sign detection and recognition system using a mix of multi cores CPU and AMD GPU is 65 frames/sec for the full image and 90 frames/sec when cropping the effective part of the image size. For future study work on the application possibility of the detection and recognition system by using smart phones or FPGA platform.

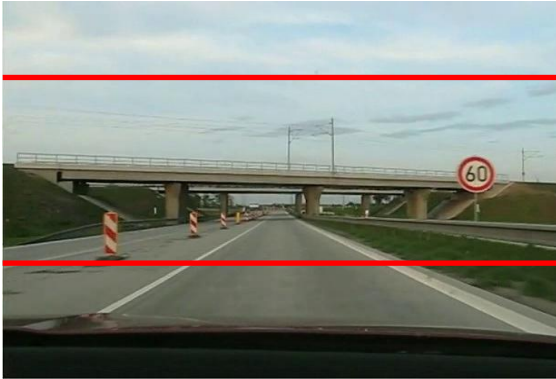


Fig (7): A video frame with the sign is in the active part of the image

6. REFERENCES

- [1] Torresen, J., J. W. Bakke and L. Sekanina., 2004. Efficient recognition of speed limit signs. In Intelligent Transportation Systems, Washington, D.C., USA. Proceedings. The 7th International IEEE Conference on, pp. 652-656.
- [2] Fleyeh, H. 2005. Traffic signs color detection and segmentation in poor light conditions. MVA2005 IAPR Conference on Machine Vision Applications, Tsukuba Science City, Japan.
- [3] Ishak K. A., 2006. A Speed limits Sign Recognition System Using Artificial Neural Network. in Research and Development, MALAYSIA, pp. 127-131.
- [4] Zakir, U., A. N. J. Leonce, and E. A. Edirisinghe, 2010. Road sign segmentation based on colour spaces: a comparative study. Proceedings of the 11th IASTED International Conference Computer Graphics and Imaging (CGIM), Innsbruck, Austria.
- [5] Thouti, K. and S. R. Sathe, 2012. Comparison of OpenMP & OpenCL parallel processing technologies. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.4.
- [6] Dore, A., and S. Lasrado, 2014. Performance analysis of Sobole edge filter on heterogeneous system using OpenCL. IJRET: International Journal of Research in Engineering and Technology, Volume: 03 Special Issue: 03, Available @ <http://www.ijret.org>.
- [7] Slabaugh, G. R. Boyes and X. Yang, . Multicore image processing with OpenMP .
- [8] Inc.(AMD), 2010. Introduction to OpenCL™ Programming. Training guide , publication #: 137-41768-10. Rev: A. Issue.
- [9] Modi, S. , 2011. Automated coin recognition system using ANN. M.Sc. thesis, Thapar University, PATIALA.
- [10] Bedi, A., 2011. A Colour segmentation method for detection of New Zealand speed signs. M.Sc. thesis, School of Engineering, Auckland University of Technology, New Zealand.
- [11] Jamil, N., T. M. T. Sembok2, and Z. Abu Bakar1, 2008. Noise removal and enhancement of binary images using morphological operations. 978-1-4244-2328-6/08/\$25.00 © IEEE.
- [12] Paul, B., 2011. Informatics simulation & exploration of mobile license plate detection employing infrared, Canny edge detection, binary threshold and control detection for submission in limited light conditions. M.Sc. thesis, Massey University, Albany, New Zealand.
- [13] Yuen H.K., J. Princen, J. Illingworth and J. Kittler A. 1989. Comparative study of Hough transforms methods for circle finding. AVC doi: 10.5244/C.3.29.
- [14] Burke, R. L. Cramer, N. Dupes, and D. McDannald, 2014. Autonomous robotic boat platform. Presentation in Department of Electrical and Computer Engineering, Bradley University.
- [15] Zhang, T. Y. and C. Y. Suen, 1984. A fast parallel algorithm for thinning digital patterns. Research Contributions, Communications of the ACM M, Volume 27 Number 3.
- [16] Damavandi, Y. and K. Mohammadi. 2004. Speed limit traffic sign detection and recognition. Cybernetics and Intelligent Systems, IEEE Conference on, 2:797{802}.
- [17] <http://techreport.com/review/23179/review-nvidia-geforce-gt-640-graphics-card>.
- [18] <http://www.notebookcheck.net/Intel-HD-Graphics4000.-69168.0.html>
- [19] Ravishekhar Banger, B. and K. Bhattacharyya, 2013. OpenCL Programming by Example. Packt Publishing Ltd. Birmingham B3 2PB, UK.