

Road Lane Detection using Hough Transform Method (Daylight Condition)

H. Tia Amelia, Agus Virgono and Randy Erfa Saputra
Department of Computer Engineering, Faculty of Electrical Engineering,
Telkom University, Bandung, Indonesia

Abstract: The development of technology in the world of transportation is so rapid and autonomous driving is one of many technological developments in the field of transportation. This system requires several aspects of support, one of which is the path detection that are useful to keep the driver on the right track. Therefore, we can use Hough transform as the method for lane detection with the help of Canny edge detection which is used as the first step of path detection.

Key words: Lane detection, Hough transform, Canny edge detection, transportation, aspects of support, right track

INTRODUCTION

The development of technology in the world of transportation is growing very rapidly. Transportation was first discovered in 3500 SM, then in 1769 the first car steered by the steam is found, until eventually began to stand industrial factory engaged in transportation and system development in it (Mariut *et al.*, 2012). In line with technological developments, the future of the vehicle will be integrated with an accurate and always up-to-date navigation system. In context of autonomous driving, the ultimate goal of situational awareness is to assist in informative decision making (Mathibela *et al.*, 2015). This is not separated from the supporting aspects of the autonomous driving system (Srivastava *et al.*, 2014). Lane detection is an important component of intelligent transportation system (Kim, 2008). In 1980, the approach used in the transportation was a camera where the camera was used to detect road markings in the form of lines on the road surface as well as the detection of paths that were passed through (Borkar *et al.*, 2009). Lane detection is one of the important supporting aspects for future technological developments for path tracking systems and it is an integral part of autonomous vehicle control system (Panwar and Raut, 2015). Lane detection has significant challenges such as bad quality line, shadows of trees or the existence of another vehicles. Research on lane detection has several methods to support the system, one of them is Hough transform for feature extraction. Hough transform will be aided by several stages of pre-processing before Hough transform declares a line. Edge detection is some kind of one of them, it is Canny edge detector that used in this research. Recent lane

detection systems are still few that use an embedded system but using a PC. The methods used should be those that have detection speed, high accuracy and can be easily implemented in embedded system such as Raspberry Pi (Bilgin and Robila, 2016). Therefore, this research built a road marking detection system that is useful to help the driver on the line marking flanking the two sides of the vehicle. Detection system is built using the Hough transform method in an embedded system. Hough transform is an image transformation technique that can be used to isolate or in other words acquire features of an image which can detect straight lines. The system is expected to be a driving assistant and supports autonomous system on intelligent transportation system.

MATERIALS AND METHODS

Lane detection model: A road marker is a sign located on surface that includes a longitudinal, transverse and oblique mark or line that aims to provide direction for traffic and boundaries on the road through which it passes. The common method used in lane detection is by placing a fixed camera on the vehicle while its images are interpreted to extract meaningful information such as positions, road markings, road boundaries and direction of vehicle's heading (Wang *et al.*, 2004). Retrieval process of the image on the camera is done each frame, then converted into a grayscale and binary at last. Filters plays an important role in the process of lane detection with the aim of eliminating noise to produce good edge detection by using filter and edge detector such as Canny edge detector with automatic thresholding to obtain the edges.

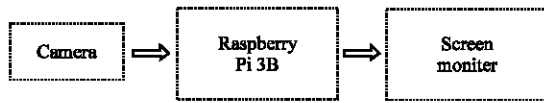


Fig. 1: Block diagram of system

The last but not least is sending the image to the lane detector after detecting the edges that will produce the left and right border. The process uses information on the edge image previously scanned by Hough transform and then Raspberry Pi compiling all the received data (Mandlik and Deshmukh, 2016).

The block diagram in Fig. 1 describes the general description of a research system that detects the moving object lines caught by the camera aimed at supporting the autonomous driving aspect. This illustrates that the camera has a function to capture every image and object. The camera will also be a means of line mark detection which is the core of this research. Raspberry Pi serves as a single board that manages all the components in the system. Raspberry Pi is assigned to perform image processing from the camera in detecting the line marking done by the Hough transform method. This monitor screen is the output of this research system which will show detected markers on the path captured by the camera that will be forwarded to the user to provide information whether the user is on the right track or not.

The flowchart in Fig. 2 shows the process of algorithm on lane detection. Figure 3-5 until Fig. 6 is a process for lane detection. Figure 3a is the input image. Figure 3b is an image that has been filtered from Fig. 3a. In Fig. 4a, the image is converted into grayscale, then the image is filtered again to the binary on Fig. 4b. Figure 5a represents method and Fig. 5b is edge detected in the image with the help Canny edge detector. Figure 6a shows the result from Fig. 5b and 6b is output image represented. This is done to find the path caught on camera.

Grayscale: The image of grayscale can be regarded as the gray value of an image. The pixel value of the input color image and the grayscale output value is in the range 0-255. The number 0 represents the black color while the 255 represents white (Kanan and Cottrell, 2012; Macedo *et al.*, 2015). The gray color level of a pixel can also be said the light level of a pixel. That is the value contained in the pixels shows the brightness of the pixels from black to white. It is usually set between 0 and 255 (for 256 gray levels) with 0 being black and 255 being white. The imagery shown in this type of image consists of 10 in gray, varying in black on the weakest intensity and white in the strongest intensity. In principle, the grayscale image is not only black and white but there is a

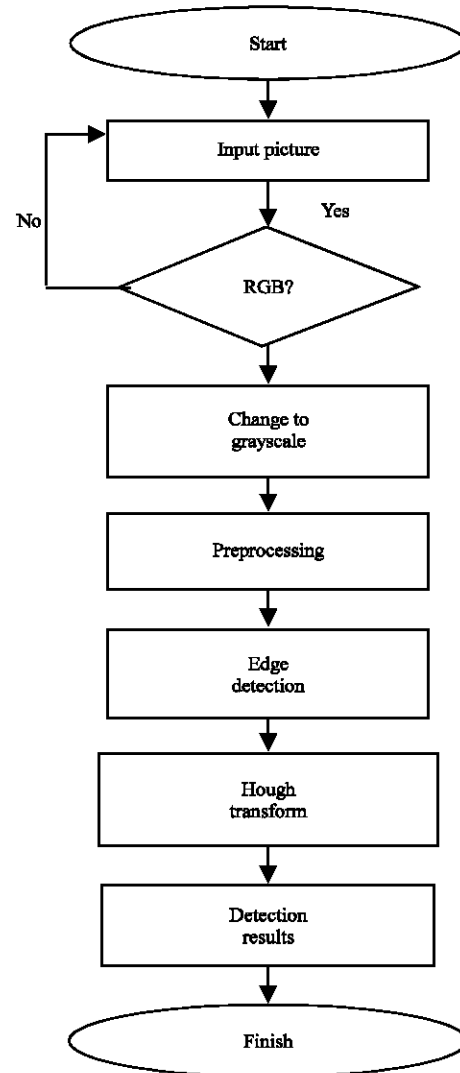


Fig. 2: Algorithm of lane detection

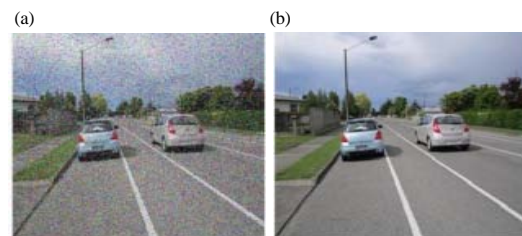


Fig. 3: a) Input image and b) Filtered image (Kaur and Kumar, 2015)

gray level in the middle. The gray degree moves from black to white. The gray scale has a range of values of $l_{min} < f < l_{max}$ or $[0, L]$ where the intensity 0 denotes black and L represents white whereas the intensity value between 0 to L shifts from black to white. Gray image can

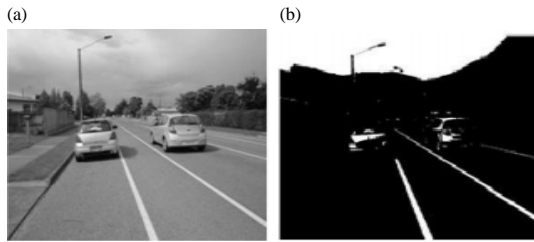


Fig. 4: a) Grayscale image and b) Binary image (Kaur and Kumar, 2015)

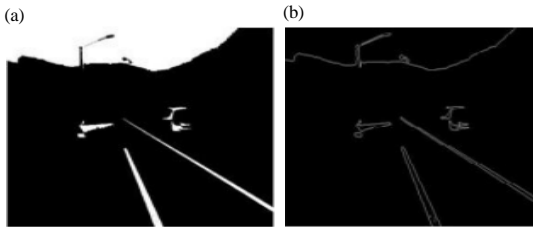


Fig. 5: a) Smoothed image and b) Edge detected image (Kaur and Kumar, 2015)

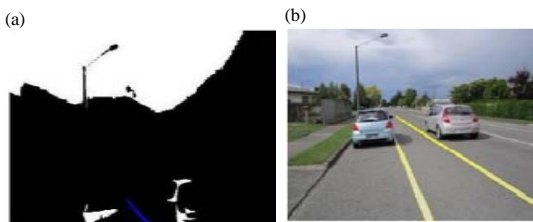


Fig. 6: a) Smoothed image and b) Output image (Kaur and Kumar, 2015)

be generated from RGB color image conversion by multiplying the three principal red, green and blue color components by a coefficient of Eq. 1 as shown as:

$$\text{Grayscale} = (R+G+B)/3 \quad (1)$$

Gray image can be generated from RGB color image conversion by multiplying all three principal red, green and blue color components by a coefficient of formula above. Figure 7 shows a 3×3 pixel value which will be converted from RGB into grayscale form. The following value of each pixel is shown in Fig. 8. From pixel value shown in Fig. 8, it can be converted into grayscale form by using Eq. 1 and yield gray value as follows:

10.6	11	12.3
10	3.6	5.3
8.6	18.3	26.6

Hough transform: Hough transform aims to directly detect lines and curves, the advantages of this method are



Fig. 7: Image with 3×3 pixels

(a)

20	12	15
12	3	10
12	15	15

(b)

10	11	5
10	5	0
2	15	10

(c)

2	10	17
8	3	6
12	25	55

Fig. 8: Value of 3×3 pixel image: a) Red; b) Green and c) Blue

less sensitive array of noise to detect a line (Song and Lyu, 2005) and high speed algorithm (Panwar and Raut, 2015). The simplest case of Hough transform is finding straight lines that are hidden in large amounts of image data. For detecting lines in image, the image is first converted into binary image using some form of thresholding (Goel, 2014). For the detection of lines with Hough transform, parametric units are used:

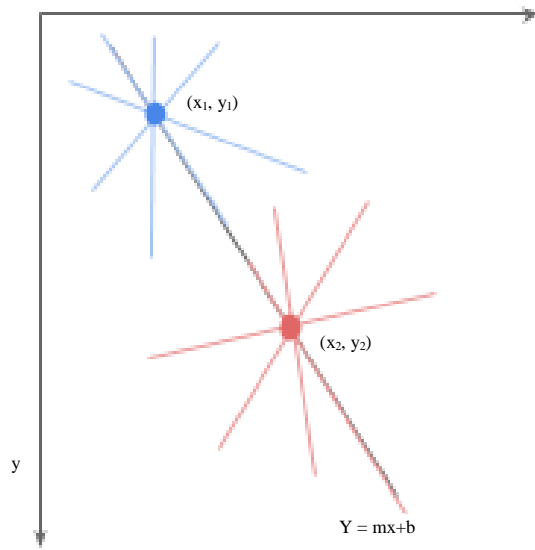


Fig. 9: Image space

$$T = x \cos d + Y \cos d \quad (2)$$

Where:

T = The radius of the origin

d = The normal angle from the origin

x, Y = The pixel line coordinates (Low *et al.*, 2014)

For example, there are two edge values (x_1, y_1) and (x_2, y_2) for each different gradient value ($m = -0.5, 1.0, 1.5$, etc.) as shown in Fig. 9 and 10. Then, calculate the corresponding b value by using the straight-line general Eq. 3:

$$y = mx + b \quad (3)$$

The lines through the edge of the image space and plots of the edge lines will be stored in the parameter space. The collinear point in the Cartesian drawing space will intersect at the point in the m - b space parameter. The Hough transformation can find the dominant lines in the image by calculating each unique equation at each point (Mariut *et al.*, 2012). The intersection in the drawing space intersects at the same point in the parameter space. The general point (m, b) that has a line in the drawing space but the slope is undefined. Therefore, the vertical line is blocked by the Hough space. The steps taken when detecting the line with the first transformation is by detecting the angle or image and the resulting binary or gray with values 0 and 1. To avoid infinity slope, we can use polar coordinate to represent a line (ρ, θ) in Hough transform method:

$$\rho = x \cos \theta + y \sin \theta \quad (4)$$

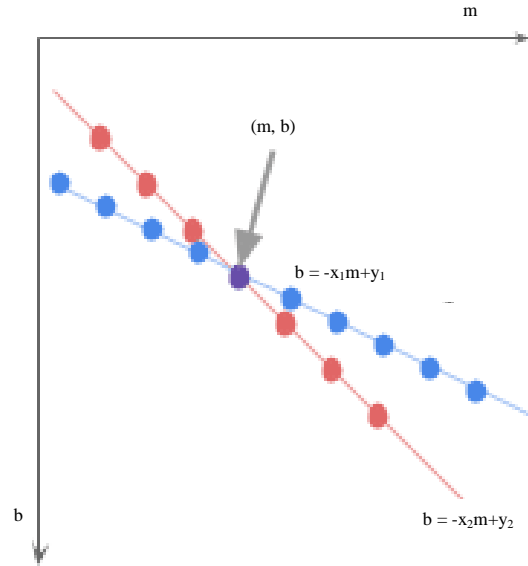


Fig. 10: Possible point that occurred



Fig. 11: Result of Canny edge detection on image

The ρ and θ grant range from $-\text{max_dist}$ to max_dist where max_dist is the length of the diagonal image input. Voting in the accumulator for each edge point and for each value, find the value of ρ and the closest index in the accumulator. Each element tells how many points or pixels provide a “voting” distance for colors with parameters (ρ, θ) .

Canny edge detection: Canny edge detection is based on computational magnitude squared. The motivation of Canny edge detection is to obtain optimal results, providing optimal detection of the actual edges and minimizing the possibility of failure. Edge detection is performed before the system determines which part is said to be a line on the mark using the Hough transform method (Aly, 2008). Canny edge detection is related to Gaussian filter at the filtering stage. After edge detection is done and get the result according to region image or area on the picture, then detection on line will be done. Figure 11 shows example of Canny edge detection result. In addition, there are several advantages of Canny edge detection (Narvekar and Karam, 2011) such as:

- Smoothing effect to remove noise
- Good localization and response

- Enhances signal to noise ratio
- Immune to noisy environment

However, there is a disadvantage such as the complicated algorithm used in Canny method makes it time-consuming and difficult to implement to achieve real-time response speed. The first step taken to determine the edge with Canny edge is to determine the gradient in the image as follow:

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

$$\theta = \tan^{-1} \frac{G_y}{G_x} \quad (6)$$

Equation 4 and 5 explain how to obtain the magnitude and direction of the gradient (Deokar and Kaushik, 2014) where, G is the magnitude and θ is the direction of the gradient. After the gradient process is complete, the next step is to provide a threshold value with two threshold values of T_1 low threshold value and T_2 high threshold value. If a value greater than T_1 is met then it will be marked and become pixel edge and other pixels connected with edge pixels and have values greater than T_2 then they are also marked as edge pixels.

RESULTS AND DISCUSSION

This chapter will discuss the experimental results with the Hough transform method of the MP4 video format with four different resolutions and two frame rates to determine the final frame rate as well as three grades of vehicle speed to determine accuracy of each image resolution and frame rate. Table 1 shows the final frame rate in each resolution.

After each image resolution and frame rate is compared and get the result with the highest frame rate is 144p and 24 fps, it will proceed to get the best accuracy value from the each image resolution. Furthermore, Table 2 shows the accuracy result within each resolution. It could be seen that the highest accuracy is obtained by 480p resolution, following decline from high resolution to low resolution and the lowest accuracy obtained by the 144p resolution. Table 3 shows that the result obtained at a speed of 80 km/h has the highest accuracy and lowest accuracy at 40 km/h.

Figure 12 and 13 illustrate the magnitude of final frame rate occurring in each resolution with frame rates of 24 fps and 30 fps. Based on the result, the final frame rate increases with decreasing resolution, so, it is affected by low resolution. If the frame rate is higher, the more real-time for system detection. Some factors that can

Table 1: Result of final frame rate vs. resolution

Resolution (p)	Final frame rate	
	(Original frame rate: 24 fps)	(Original frame rate: 30 fps)
480	2.24	1.81
360	4.04	3.26
240	8.69	6.26
144	12.59	10.72

Table 2: Result of accuracy vs. resolution

Resolution (p)	Accuracy (%)
480	100.00
360	100.00
240	98.17
144	80.00

Table 3: Result of accuracy vs. vehicle speed

Speed (km/h)	Accuracy (%)
40	80
60	89
80	86

decrease frame rate such as the number of bits per pixel in the process which is too large due to the large file size can affect the process time on the system and generate large delay. The device performance condition also becomes one of the factors of the process time in the condition of the device that is being active too long can make the hardware becomes busy and get hot that cause its performance decreased. Note the graph that shows the highest resolution results in the lowest frame rate for the detection system.

Figure 14 illustrates the value of accuracy from each resolution. Resolution affects the accuracy of a system, recorded at the highest resolution of 480p having the highest accuracy while the lowest resolution with 144p value is 80%. When resolution is higher, the image will be clearer, so, system can detect a line of marking well during the day

Figure 15 shown the accuracy at each speed and has average value 80%. At 60 km/h system has the highest accuracy of 89% and lowest accuracy is at 40 km/h with an average accuracy of 80%. It can be seen from these various speeds that if the speed is too fast, it will affect the detection accuracy about 14% of the maximum value. It is influenced by shaking experienced by the camera at the speed of 80 km/h, thus, affecting the gray value of each pixel in the image. However, at a 40 km/h speed, it has an accuracy lower than 60 and 80 km/h. This can occur due to several factors such as the faint line that resembles a line of markers with bright natural lighting conditions that make the system difficult to distinguish, so, the system unable to detect the line well. In addition, the distance of the dotted line marks on the right that are too far away from each other captured by the camera becomes one of the factors that results an undetectable line. It is because the Hough transform method works by scanning point by point from a line.

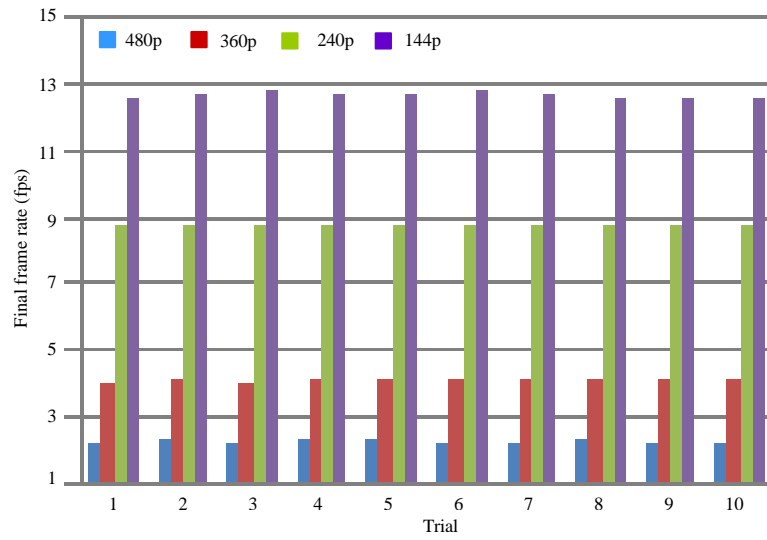


Fig. 12: Comparison of final frame rate in various video resolution with 24 fps original frame rate

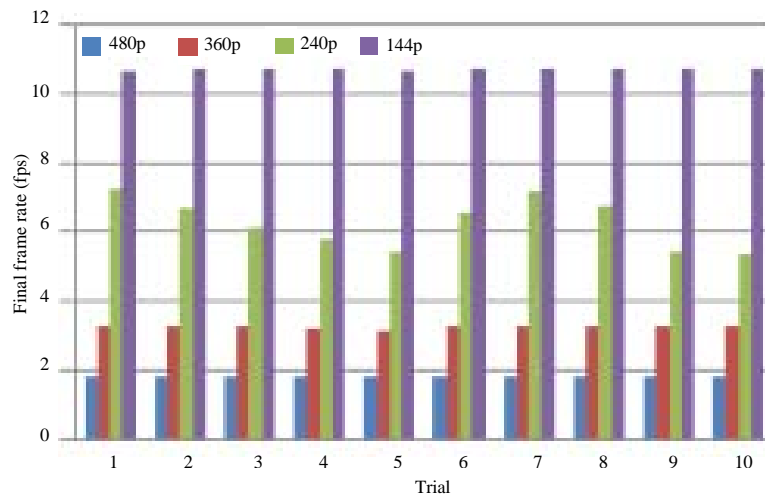


Fig. 13: Comparison of final frame rate in various video resolution with 30 fps original frame rate

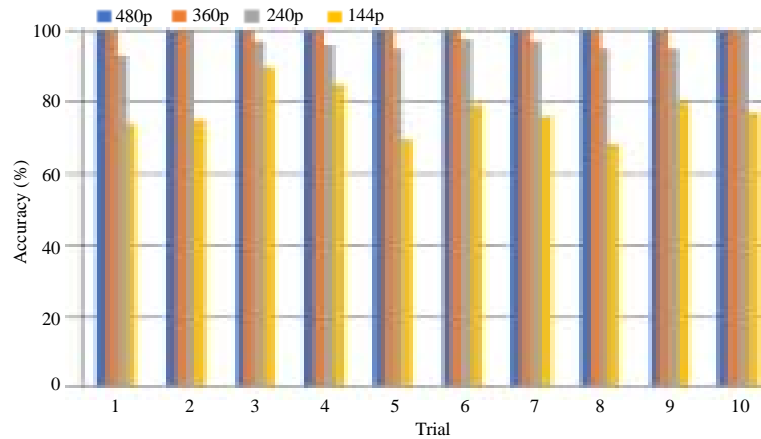


Fig. 14: Comparison of detection accuracy in various video resolution

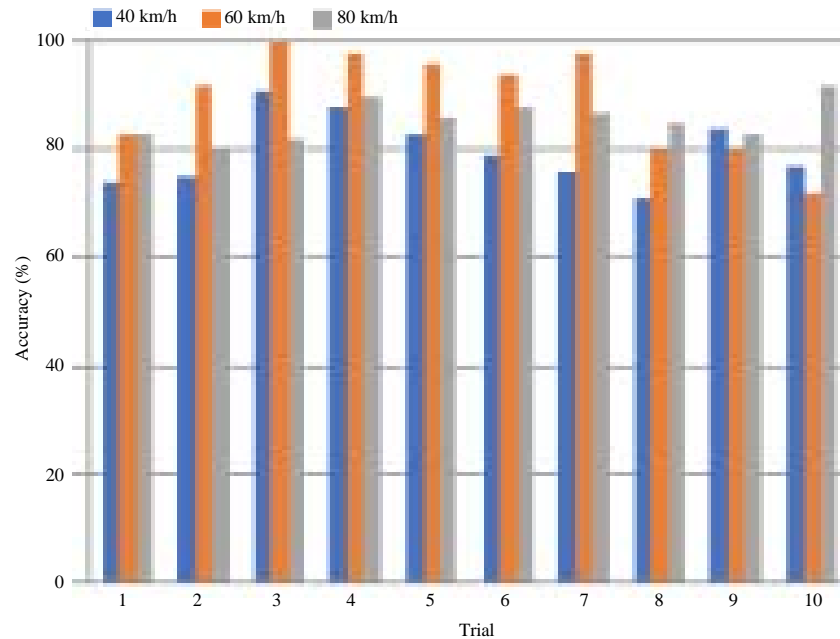


Fig. 15: Comparison of detection accuracy in various vehicle speed

CONCLUSION

Based on the experiment results using the Hough transform method that has been obtained, it can be concluded that:

- The minimum detection rate of lane marker using Hough transform method in daylight is 80%
- The highest frame rate that can be achieved by this system is 12.59 fps using 144p video resolution
- Video resolution that affects the detection accuracy caused by the image quality or clarity level of an image
- Unfavorable road conditions and speed of the vehicle will affect the stability of the system accuracy

REFERENCES

- Aly, M., 2008. Real time detection of lane markers in urban streets. Proceedings of the 2008 IEEE International Symposium on Intelligent Vehicles, June 4-6, 2008, IEEE, Eindhoven, Netherlands, ISBN:978-1-4244-2568-6, pp: 7-12.
- Bilgin, E. and S. Robila, 2016. Road sign recognition system on Raspberry Pi. Proceedings of the 2016 IEEE International Conference on Long Island Systems, Applications and Technology (LISAT), April 29, 2016, IEEE, Farmingdale, New York, USA., ISBN:978-1-4673-8490-2, pp: 1-5.
- Borkar, A., M. Hayes, M.T. Smith and S. Pankanti, 2009. A layered approach to robust lane detection at night. Proceedings of the IEEE 2009 International Workshop on Computational Intelligence in Vehicles and Vehicular Systems, March 30-April 2, 2009, IEEE, Nashville, Tennessee, USA., ISBN:978-1-4244-2770-3, pp: 51-57.
- Deokar, P.S. and A.R. Kaushik, 2014. Review on distributed canny edge detector using FPGA. Intl. J. Adv. Res. Electr. Electron. Instrum. Eng., 3: 11849-11856.
- Goel, A., 2014. Lane detection techniques-a review. Intl. J. Comput. Sci. Mob. Comput., 3: 596-602.
- Kanan, C. and G.W. Cottrell, 2012. Color-to-grayscale: Does the method matter in image?. PloS. One, 7: 1-7.
- Kaur, G. and D. Kumar, 2015. Lane detection techniques: A review. Intl. J. Comput. Appl., 112: 4-8.
- Kim, Z., 2008. Robust lane detection and tracking in challenging scenarios. IEEE Trans, Intell. Transp. Syst., 9: 16-26.
- Low, C.Y., H. Zamzuri and S.A. Mazlan, 2014. Simple robust road lane detection algorithm. Proceedings of the 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS), June 3-5, 2014, IEEE, Kuala Lumpur, Malaysia, ISBN:978-1-4799-4653-2, pp: 1-4.

- Macedo, S., G. Melo and J. Kelner, 2015. A comparative study of grayscale conversion techniques applied to SIFT descriptors. *SBC. J. Inter. Syst.*, 6: 30-36.
- Mandlik, P.T. and A.B. Deshmukh, 2016. Raspberry-Pi based real time lane departure warning system using image processing. *Intl. J. Eng. Res. Technol.*, 5: 762-766.
- Mariut, F., C. Fosala and D. Petrisor, 2012. Lane mark detection using hough transform. *Proceedings of the 2012 International Conference and Exposition on Electrical and Power Engineering*, October 25-27, 2012, IEEE, Iasi, Romania, ISBN:978-1-4673-1173-1, pp: 871-875.
- Mathibela, B., P. Newman and I. Posner, 2015. Reading the road: Road marking classification and interpretation. *IEEE. Trans. Intell. Transp. Syst.*, 16: 2072-2081.
- Narvekar, N.D. and L.J. Karam, 2011. A no-reference image blur metric based on the cumulative probability of blur detection (CPBD). *IEEE. Trans. Image Proc.*, 20: 2678-2683.
- Panwar, S. and S. Raut, 2015. Survey on lane detection using Hough transform technique. *Intl. J. Adv. Res. Electr. Electron. Instrum. Eng.*, 4: 401-405.
- Song, J. and M.R. Lyu, 2005. Hough transform based line recognition method utilizing both parameter space and image space. *Pattern Recognit.*, 38: 539-552.
- Srivastava, S., R. Singal and M. Lumba, 2014. Efficient lane detection algorithm using different filtering techniques. *Intl. J. Comput. Appl.*, 88: 6-11.
- Wang, Y., E.K. Teoh and D.G. Shen, 2004. Lane detection and tracking using B-Snake. *Image Vision Comput.*, 22: 269-280.