2019 6th International Conference on Electric Vehicular Technology (ICEVT)
November 18-21, 2019, Bali, Indonesia

Real-Time Image Processing Method Using Raspberry Pi for a Car Model

Mochammad Ariyanto
Department of Mechanical Engineering
Diponegoro University
Semarang, Indonesia
National Center for Sustainable Transportation Technology, Indonesia
mochammad_ariyanto@ft.undip.ac.id

Ismoyo Haryanto
Department of Mechanical Engineering
Diponegoro University
Semarang, Indonesia
National Center for Sustainable Transportation Technology, Indonesia
ismoyo2001@yahoo.de

Joga Dharma Setiawan
Department of Mechanical Engineering
Diponegoro University
Semarang, Indonesia
National Center for Sustainable Transportation Technology, Indonesia
joga.setiawan@ft.undip.ac.id

M. Munadi
Department of Mechanical Engineering
Diponegoro University
Semarang, Indonesia
ational Center for Sustainable Transportation Technology, Indonesia
muna_096@yahoo.com

M. Sri Radityo
Department of Mechanical Engineering
Diponegoro University
Semarang, Indonesia
radityo.ladinegara@gmail.com

***Abstract*—This paper presents the development of a car model that can detect edge, line, and corner of the road image and also the model can detect the red color of a traffic light image. The car model is equipped with a camera which is used for computer vision purpose. The image comes from a camera is read by using Raspberry Pi single-board computer. The algorithms for image processing methods are selected to detect edge, line, corner, and traffic light color of the road model. The algorithms are developed in Simulink diagram block and embedded into Raspberry Pi using Simulink Support Package for Raspberry Pi Hardware. The embedded algorithms for detecting line, edge, corner and red color of traffic light will be tested. The test will be conducted in real-time mode. Based on the test results, the embedded image processing algorithms can successfully detect line, edge, and corner of the road images, and detect the red color of traffic light image.**

Keywords—image processing, Raspberry Pi, real-time, car model

I. INTRODUCTION

WHO's Global status report on road safety 2015 presents information on road safety from 180 countries. Indicates that worldwide the total number of road traffic deaths has plateaued at 1.25 million per year. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury. Road traffic injuries cause considerable economic losses to individuals, their families, and to nations as a whole. Road traffic crashes cost most countries 3% of their gross domestic product.

The vehicle safety system should be increased to reduce the number of road accidents. An autonomous vehicle is a technology that has a great role in improving the safety of driver. The first step that we used to create autonomous vehicle is image processing and computer vision in order to car model recognize and tracking the white lane markers of the roadway and can detect traffic light intersection, if red color is on, the car must stop.

A lot of research in lane detection and recognition use Phyton under OpenCV software. The widely used sensor is a camera [1]–[8], but some of utilized sensors are RFID [9]. The camera is used to acquire image or video. The image then is processed using image processing and pattern recognition method for detection and pattern recognition. The image processing method is used for lane detection and traffic light detection. Ref [6] used color extraction to detect boundary lines of the lane under various lighting conditions. Y. Wang [7] proposed a Catmull–Rom spline-based lane model which describes the perspective effect of parallel lines [10].

Recently, the research of the lane and traffic detection have grown increasingly. Many of them used Raspberry Pi single-board computer for low-cost embedded systems [8], [11]–[13]. For engineers who do not have knowledge in programming especially in python or C language, they can use Matlab/Simulink for image processing method [14]. A lot of research uses Raspberry Pi for signal processing, image processing, and in robotic applications.

In this paper, the car model is equipped with a low-cost Raspberry Pi single-board computer and a camera. Image processing algorithm for lane detection and traffic light detection will be developed under MATLAB/Simulink environment. The image processing method is aimed to detect line, edge, corner of the road image, and it can detect the red color of traffic light image. In future research, after developing the image processing method, it will be implemented in a car model, and it will be controlled using artificial intelligence.

A. Car Model

The modified car model is powered by Dynamo N20 DC 6-12V 300 RPM Gear Motor High Torque. This car model uses one micro servo motor to control the steering angle of the car model. This car model employs Arduino Nano to control its speed and steering angle by giving the command to the DC motor and micro servo motor. A selected functional car model can be seen in Fig. 1. The car model is equipped with camera module and Raspberry Pi for real-time image processing.

978-1-7281-2917-4/19/$31.00 ©2019 IEEE

Fig. 1. Modified car model equipped with Raspberry Pi and camera module (R/C 1:10).

B. Raspberry Pi 2 Model B and Raspi Camera

The Raspberry Pi 2 Model B is the second-generation Raspberry Pi. It replaced the original Raspberry Pi 1 Model B+ in February 2015. The Raspberry Pi 2 Model B (Raspi) originally intended for educational purposes has become famous immediately after its introduction in 2012. The Raspberry Pi 2 delivers 6 times the processing capacity of previous models. This second-generation Raspberry Pi has an upgraded Broadcom BCM2836 processor, which is a powerful ARM Cortex-A7 based quad-core processor that runs at 900MHz. The board also features an increase in memory capacity to 1Gbyte.

For computer vision purposes, the Raspberry Pi Camera rev 1.3 Board (Raspi Camera) plugs directly into the CSI connector on the Raspberry Pi. The Raspberry Pi Camera Board plugs directly into the CSI connector on the Raspberry Pi. It's able to deliver a crystal clear 5 MP resolution image, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p/30fps, 720p/60fps, and 640x480p 60/90 video recording.

Raspi Camera is custom designed and manufactured by the Raspberry Pi Foundation in the UK. The Raspberry Pi Camera Board features a 5 MP Omni vision 5647 sensor in a fixed-focus module. The module attaches to Raspberry Pi, by way of a 15 Pin Ribbon Cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor.

The board itself is tiny, at around 25x20x9 mm, and weighs just over 3g, making it perfect for autonomous car model applications where size and weight are important. The proposed of single-board computer and the camera can be seen in Fig. 2.

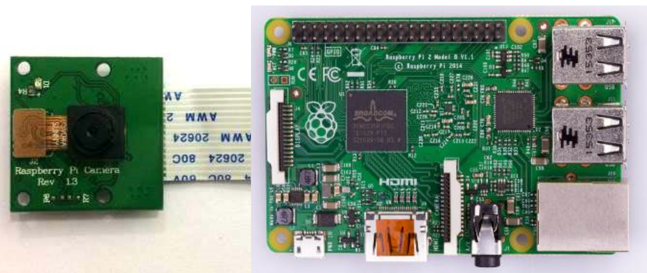

Fig. 2. The Raspberry Pi Camera Rev 1.3 board (left) and Raspberry Pi 2 Model B V1.1 (right).

II. IMAGE PROCESSING METHOD

The proposed method uses Raspberry Pi board as the main computation for real-time image processing. Using real-time image processing methods, many functions are performed such as line detection, edge detection, corner detection and color of traffic light detection for detection of the road condition. In this study, the used image of road model and the car model can be depicted in Fig. 3. The previous study of image processing method can be seen in [1], [8], [10], [15].

Fig. 3. Original Image of the car model and the road line model.

A. Line Detection and Recognition

Lanes in the roadway are normally marked by white segments, which can be identified and captured by real-time vision system's optical sensor (RPi Cam) in RGB format. The usefulness of road line readings is to keep the car's position on track when touching the road line, then the car will automatically get back on the track.

In this study, the used method in line detection is Hough Transform. Hough Transform is a very often used method for reading patterns such as line, circle, ellipse, and parabola. The advantage of using Hough Transform is to detect an edge with gap at the feature boundary and is relatively unaffected by noise.

The general equation for finding the line is as follows:

$$x\cos(\theta) + y\sin(\theta) = r \tag{1}$$

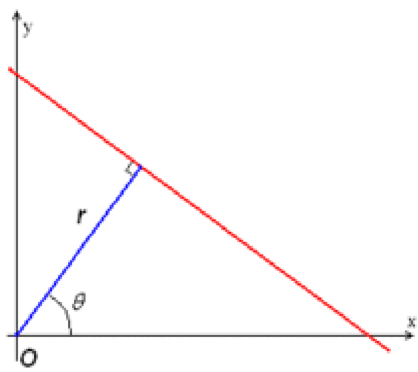

Fig. 4. The coordinate space of x and y for finding the line.

B. Edge Detection

Edge detection is an operation performed to detect borders that constrain two homogeneous imagery areas that have different brightness levels. The goal of edge detection is to convert the 2D image into a curve shape. Edge is some part of the image where the intensity of the brightness changes drastically.

There are several methods in edge detection, among them are Robert method, Prewitt method, Sobel method, and others. The method used in this experiment is the Sobel method. The Sobel method has the ability to reduce noise before performing edge detection calculations so that the edge is generated more than before. This method uses two kernels of 3x3 pixels for the gradient calculation.

The gradient quantities calculated using a Sobel operator are as follows:

$$G = \sqrt{S_x^{\,2} + S_y^{\,2}} \tag{2}$$

where

G is for = Gradient of Sobel Operator

Sx=Horizontal Sobel Gradient

S_y=Vertical Sobel Gradient

G is also the gradient at the midpoint of the kernel and the partial derivatives that are calculated using the following equation as expressed in equation (3) and (4).

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \tag{3}$$

$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \tag{4}$$

Where c is a constant value of 2. S_y and S_x are implemented into the following kernels:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{5}$$

$$S_y = \begin{bmatrix} -1 & 2 & 1 \\ -2 & 0 & 2 \\ -1 & -2 & -1 \end{bmatrix} \tag{6}$$

C. Corner Detection

Corner detection is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, panorama stitching, 3D modeling and object recognition. Moreover, corner detection algorithm is one of the earliest corner detection algorithms and defines a corner to be a point with low self-similarity [10].

There is more sophisticated corner detectors named after their authors. Harris and Stephens algorithm is selected in detecting the corner of the line of road image. The Harris and Stephens formula can be found in ref. [16], [17]. Harris and Stephens Corner Detection is a corner detection algorithm that is often used because it can produce a consistent value in the image that experiencing the rotation, scaling, different light intensity conditions or have a lot of noise in the image [8].

D. Traffic Light Detection

To reduce the accident that happened at the traffic light intersection, we develop Raspi camera based on algorithm which is real-time to detect red light of traffic light. We use simple color extraction method. This is useful for controlling the vehicle to stop if the red traffic light when it is turned on. The color detection of traffic light has been studied previously in ref [18]. The original traffic light image for red light detection can be seen in Fig. 5. In this method, we use the color threshold for converting the RGB image to binary image. The threshold is selected based on the red color. The proposed method will be tested in Fig. 5 and the resulting image is presented in section III.

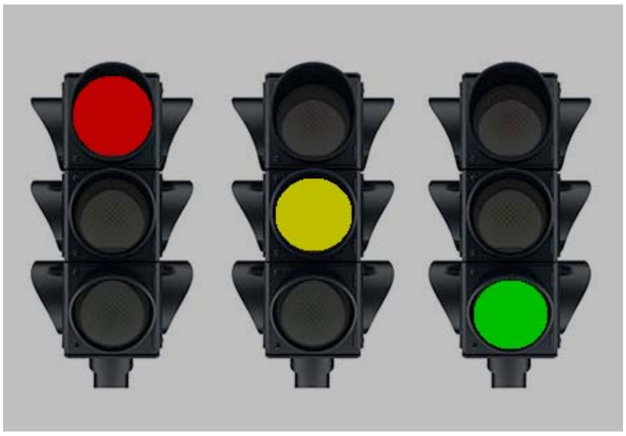

Fig. 5. The original image of Traffic Light Detection.

III. Implementation in MATLAB/Simulink

MATLAB is an interactive program for numerical computation and data visualization. Simulink is a MATLAB graphical extension to model and simulate a system. In Simulink, the system is described as a block diagram, including transfer function, summing junction, in which there are also virtual input and output devices. While in Simulink, data/information from various blocks sent to other blocks connected with line. The user without programming

In this study, the image processing algorithms discussed in section III will be developed under MATLAB/Simulink environment. The image processing block uses Simulink Support Package for Raspberry Pi Hardware. it can be downloaded freely on the MathWorks web. For reading the image/video from Raspi camera, video capture block is used. While the image processing algorithm uses Computer

Vision System Toolbox.

The diagram block for line detection can be seen in Fig. 6. The image/video is captured using Video Capture block. The image is converted to intensity image using color space conversion. It processed to edge detection using Sobel operator and then it processed using Hough transform and Hough line.

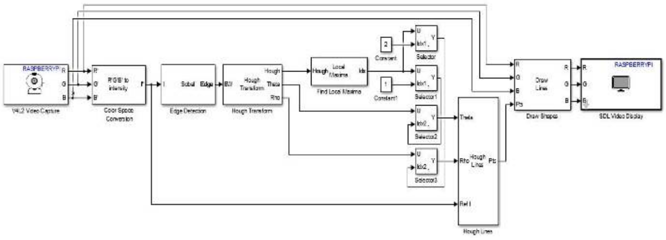

Fig. 6. Block diagram of line detection in the Simulink block diagram.

For the corner detection, image/video is captured using Video Capture block. The captured image/video is converted into an intensity image. The image then processed using corner detection method block using Harris & Stephen as shown in Fig. 7. For edge detection, image/video is captured using Video Capture block. The captured image/video is converted into intensity image. The image then processed by edge detection block using Sobel operator as shown in Fig. 8. In red light detection, the algorithm uses color selection using threshold. The threshold is selected based on the red color as shown in Fig. 9.

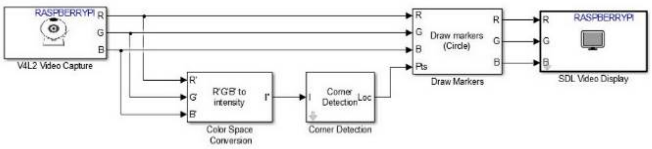

Fig. 7. Block diagram of corner detection in Simulink block diagram.

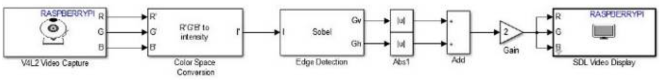

Fig. 8. Block diagram of edge detection in Simulink block diagram.

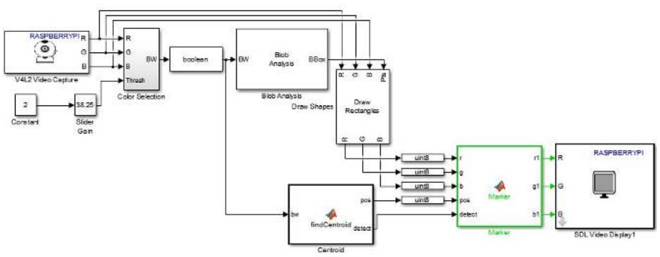

Fig. 9. Block diagram of traffic light detection in Simulink block diagram for red.

IV. Experimental Results

Real-time image processing for line detection, edge detection, corner detection and red light of traffic light detection used Raspberry Pi 2 Model B with Raspberry Camera. The proposed methods from Fig. 6 to Fig. 9 are embedded to Raspberry Pi single-board computer using Simulink Support Package for Raspberry Pi Hardware. The communication of host computer and Raspberry Pi is conducted by using ethernet connection. The selected pixel is 640x480. The chosen frame per second (fps) in this research is 30 fps. This is the highest frame per second that can be used in image/video processing using Raspberry Pi. The line and traffic light detection are tested in indoor room with fixed lighting.

A. Line Detection

Line detection of RGB image is conducted after the RGB image is converted to an intensity image. The process of converting the RGB image to intensity is conducted using Color space conversion block. Intensity image can be represented as a single matrix. Each element of the matrix in intensity image corresponding to one image pixel. The resulted from line detection is shown in Fig. 10. Based on the figure, it can find the right lane on the road image and detect line on the right side of the road image. The detected line is indicated with blue line. Based on the resulted image, the proposed method cannot detect dashed line and line having corner. It can detect perfectly on the continuous line.

B. Edge Detection

The initial image for edge detection is RGB image. The image must be converted into an intensity image. The resulted image from edge detection is shown in Fig. 11. The edge detection using Sobel operator can detect the straight double lines on the left side and right side, as well as dashed line in the center of the road. The edge detection detects all of the edges in Fig. 10 including the car model image.

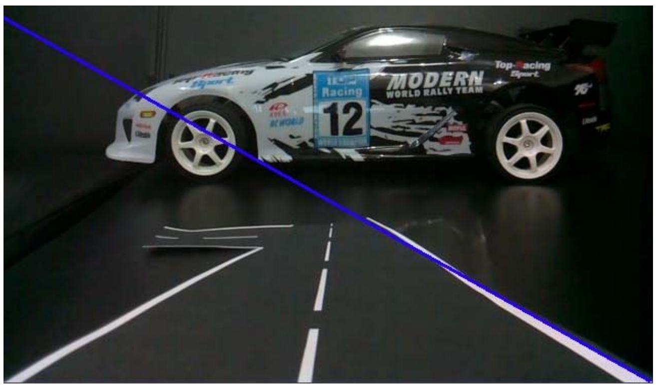

Fig. 10. Resulted line detection.

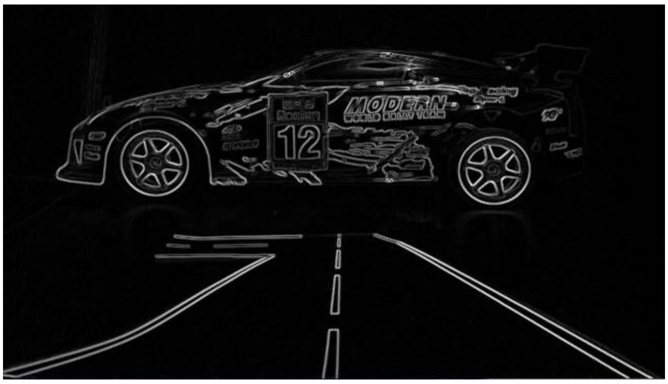

Fig. 11. . Edge Detection by Sobel Operator.

C. Corner Detection

The performance of corner detection using the road image can be depicted in Fig. 12. The Harris and Stephens corner detection can detect the left corner of the road image. The detected left corner on the road image can be indicated with a green mark as shown in Fig. 12. Based on the image, the method can detect sharp corner with the angle of 90 degrees. The method can also detect the tip of the line road of the image.

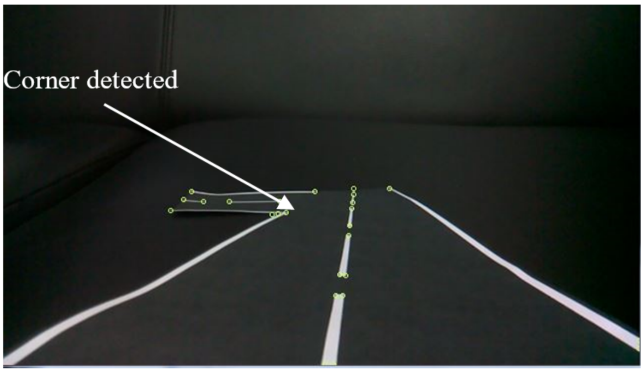

Fig. 12. . Result of corner detection image using Harris & Stephen.

D. Traffic Light Detection

The developed algorithm can detect the red light on the traffic light as shown in Fig. 13. The detected red light is indicated by red square line with a circle green mark on its center. When the yellow and green are light on, the image processing method does not detect the colors.

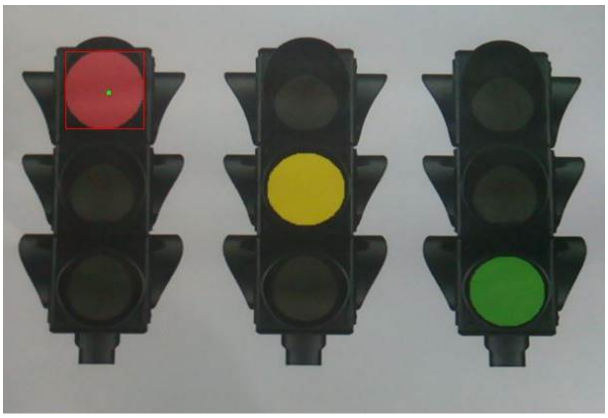

Fig. 13. . Red Color Traffic Light Detection.

V. Conclusion and Future Research

Based on the results in section IV, the line detection method can be implemented for an autonomous car model that use camera as vision sensor. The line detection will be used in detection of road images when the road is straight line. The Harris and Stephens corner detection can be implemented for autonomous car model for detecting the corner of the road. For edge detection, it is hard to implement the method for an autonomous car because it will detect all of the edges captured by camera. Red color detection of traffic light still has the challenge to implement it on the outdoor. Because the proposed method is not robust to the various lighting condition.

In future research, the corner detection will be enhanced for estimating the angle of the line from o degree to 90 degrees. The traffic light detection will be improved using pattern recognition method with neural network. The method will be tested under various lighting conditions.

Acknowledgment

The authors gratefully acknowledge partial funding support from the United States Agency International Development (USAID) under the Sustainable Higher Education Research (SHERA) program and research fund from the Faculty of Engineering, Diponegoro University.

References

[1] G. Mu, Z. Xinyu, L. Deyi, Z. Tianlei, and A. Lifeng, (2015), Traffic light detection and recognition for autonomous vehicles, *J. China Univ. Posts Telecommun.*, vol. 22, no. 1, pp. 50–56.

[2] M. Omachi and S. Omachi, (2009), Traffic light detection with color and edge information, in *2009 2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 284–287.

[3] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, (2016), Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives, *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1800–1815.

[4] K.-H. Lu, C.-M. Wang, and S.-Y. Chen, (2008), Traffic light recognition," *J. Chin. Inst. Eng.*, vol. 31, no. 6, pp. 1069–1075.

[5] N. Fairfield and C. Urmson, (2011), Traffic light mapping and detection," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 5421–5426.

[6] H. Y. Cheng, B. S. Jeng, P. T. Tseng, and K. C. Fan, Lane Detection With Moving Vehicles in the Traffic Scenes, *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571–582, Dec. 2006.

[7] Yue W, Dinggang S, and Eam K. T, Lane detection using spline model, *Pattern Recognit. Lett.*, vol. 21, no. 8, pp. 677–689, Jul. 2000.

[8] K. Horak and L. Zalud, (2016), Image Processing on Raspberry Pi for Mobile Robotics, *Int. J. Signal Process. Syst.*, pp. 494–498.

[9] A. Ali and H. A. Hussein, (2017), Traffic lights system based on RFID for autonomous driving vehicle, in *2017 Annual Conference on New Trends in Information Communications Technology Applications (NTICT)*, pp. 122–127.

[10] Muhammad S, Muhammad F, and Prashant P, (2013), Robust lane detection and object tracking. Sweden: Master's Thesis, School of Engineering, Blekinge Institute of Technology.

[11] M. Malathi and R. Geetha, "A Real Time Image Processing Based Fire Safety Intensive Automatic Assistance System using Raspberry Pi," *Int. J. Mod. Trends Sci. Technol.*, pp. 34–38, 2016.

[12] Vikas Reddy Sudini, *Real-time Vision-Based Lane Detection with 1D Haar Wavelet Transform on Raspberry Pi*. Master Thesis, Utah State University, 2017.

[13] L. Kalms, J. Rettkowski, M. Hamme, and D. Göhringer, “Robust lane recognition for autonomous driving,” in *2017 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2017, pp. 1–6.

[14] Peter Corke, *Robotics, Vision and Control - Fundamental Algorithms in MATLAB*. Springer, 2011.

[15] M. Malathi, and R. Geetha, “A Real Time Image Processing Based Fire Safety Intensive Automatic Assistance System using Raspberry Pi,” *Int. J. Mod. Trends Sci. Technol.*, vol. 2, pp. 34–38, 2016.

[16] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[17] Y. Xiao, Y. Wang, X. Dan, A. Huang, Y. Hu, and L. Yang, “Corner detection and sorting method based on improved Harris algorithm in camera calibration,” presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2016, vol. 0023.

[18] T. H. P. Tran, C. C. Pham, T. P. Nguyen, T. T. Duong, and J. W. Jeon, “Real-time traffic light detection using color density,” in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2016, pp. 1–4.