Journal of Engineering and Applied Sciences 14 (20): 7495-7501, 2019
ISSN: 1816-949X

Speed Sign Detection and Recognition using Histogram of Oriented Gradient and Support Vector Machine Method on Raspberry Pi

Yosua Pangihutan Sagala, Agus Virgono and Randy Erfa Saputra
Department of Computer Engineering, Faculty of Electrical Engineering,
Telkom University, Bandung, Indonesia

Abstract: Advance Driving Assistance System (ADAS) as a standard safety feature in modern vehicles is one of the most developed transportation technologies. The ADAS itself is built by several subsystems, one of which is the detection and recognition of traffic signs. This study presents a system of detection and recognition of the speed limit traffic signs on the roadside with certain conditions. The process of detecting traffic signs using HOG (Histogram of Oriented Gradient) as a feature of image and classified them using SVM (Support Vector Machine) method. With the detection and recognition system of traffic signs, it is expected to improve the component of ADAS. The output of this system is information about the allowed speed limits on the road based on detected and recognizable sign. Test result shows the system yields accuracy more than 80% for detection and recognition.

Key words: Traffic sign detection, histogram of oriented gradient, support vector machine, ADAS, HOG, recognizable

INTRODUCTION

Driving assistance technology is an important component of modern vehicles (Ardianto *et al*., 2017). ADAS (Advance Driving Assistant System) can recognize the traffic environment around the vehicle and provide the navigation system automatically. One of the important technologies used in ADAS is the detection and recognition of traffic signs (Ellahyani and Ansari, 2016; Zhang *et al*., 2014). Many methods are used in research on the detection and recognition of traffic signs. Each algorithm and method has its advantages and disadvantages and its use depends on the requirement of the system that wants to be built. In the detection and recognition process, Genetic algorithm, Hough transform, shaped based algorithm, color-based detection whereas in the process of classification is OCR (Optical Character Recognition), k-nearest neighbor, euclidian distance, HOG (Histogram of Oriented Gradient) and SVM (Support Machine Vector) are used commonly in many research (Bilgin and Robila, 2016; Sugiharto and Harjoko, 2016). This study presents a system of the detection and recognition of speed limit traffic sign using Raspberry Pi device that can process the image by applying HOG (Histogram of Oriented Gradient) method and SVM (Support Machine Vector) from image that have been captured by the camera and recognize traffic speed limit signs. Therefore, it can be a support system of intelligent transportation in detecting traffic speed limit sign on the roadside and provide information about the allowed speed limit on the road.

MATERIALS AND METHODS

System design: Detection and recognition system of traffic sign has several stages in the process as shown in Fig. 1. The system built starts with capturing image. Then, the captured image go into the process of detection of signs which used histogram of oriented gradient method to extract the features of the image and the feature will be classified using the support vector machine method. If the obtained image detected as a traffic sign, the object will be focused in one ROI (Region of Interest) for the next stage which is the recognition phase of traffic signs. Detected sign from the previous stage then used in the recognition process by using the support vector machine method. In this method, previous dataset of road sign images with JPG format has been created and labeled manually in a folder which contains speed limit sign of 80, 40 and 20 km/h. The system will classify the detected traffic

Corresponding Author: Yosua Pangihutan Sagala, Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

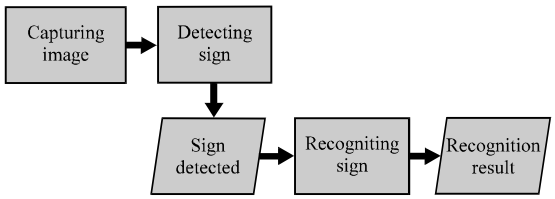

Fig. 1: Block diagram of system

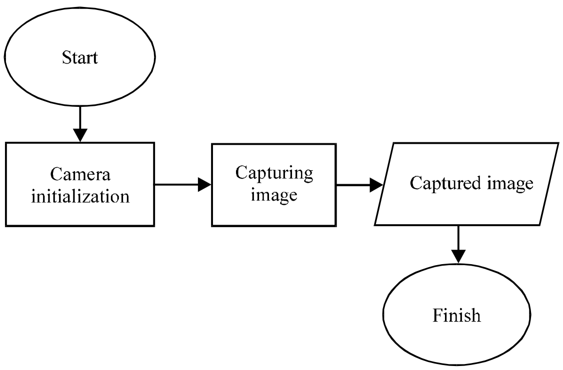

Fig. 2: Image acquisition process

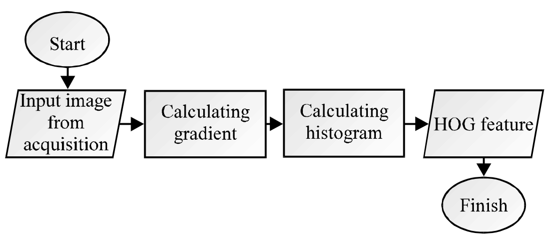

Fig. 3: Feature extraction process

signs based on training data and the output will be displayed as text information from the traffic sign image in the monitor screen.

Image acquisition: The image that will be inputted into the system is captured using Raspberry Pi camera module. The image processed on the system has a PNG format. As shown in Fig. 2, before the camera captures the input image, there is an initialization process from the camera. This process aims to set the resolution size and color space of the image before capturing.

Feature extraction: Figure 3 shows the feature extraction process using histogram of oriented gradient method. This stage aims to get important information from an image which is HOG feature. The image that has been obtained previously in the acquisition process is a true color image which is a representation of a color image that has three color components: Red, Green and Blue (RGB) and each component has 256 possible values. Before proceeding to the process of calculating the image gradient value, we need to change from RGB to grayscale. It aims to make it easier to calculate the gradient of the image because the conversion of RGB to grayscale change the pixel value which initially has 3 values of R, G and B into one value that is grayscale. The gradient calculation of the image aims to obtain the characteristics of the traffic sign image. The gradient of an image is obtained by finding the edge of the line between one region with the another region which have a high differences. After the gradient value is obtained, proceed to making histogram but in this process the image will be divided into several cells and grouped into a block which is called HOG feature. In this study, the cells in HOG are rectangular (R-HOG) and have three parameters number of cells per block, number of pixels per cell and number of bin per histogram (Kim and Kwon, 2015).

Image classification: The result of the HOG feature is converted into a vector feature and processed in the classification stage by using support vector machine method to detect and recognize the traffic sign. The purpose of SVM method is to train the data and get the hyperlane to divide field of the +1 class that contains the positive vector feature and -1 class that contains the negative vector feature (Santosa, 2010). In this study, there are two kinds of SVM method implementation the first is SVM for training and the second is SVM for testing. Figure 4 shows the steps of SVM to train the data. The data train is obtained from data collection that has been done before. The result of this learning produced a model that used as a reference to detect traffic signs in the testing process. After obtaining the SVM Model, the next step is the testing phase as shown in Fig. 5. In this stage, the camera sensor takes the image of traffic signs. The detection process is done by doing a filter from the top left corner to the bottom right corner. The sliding window size used to filter adjusted, according to the size of the training data that have been obtained. If the process found an image section corresponding to the model then the object passed by the sliding window is a traffic sign object and it will be marked with a certain colored box on the object as shown in Fig. 6.

Sign recognition process: Figure 7 shows the process of recognizing traffic signs. The recognition sub-system aims to recognize traffic signs that have been detected at the detection stage. In this sub-system, the detected sign image will go through the pre-processing stage, convert RGB image to binary image. Furthermore, the image will be classified by testing it using SVM (Support Vector

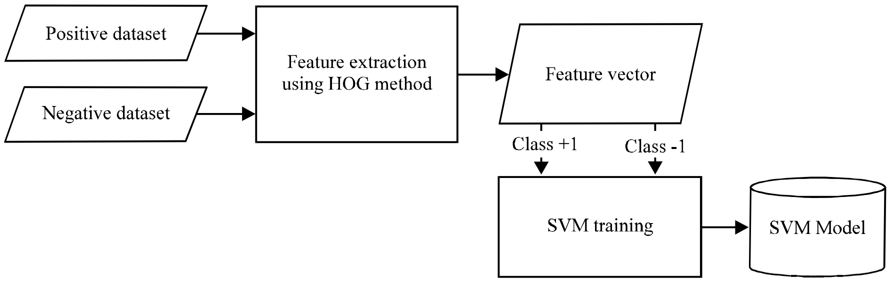

Fig. 4: SVM training process

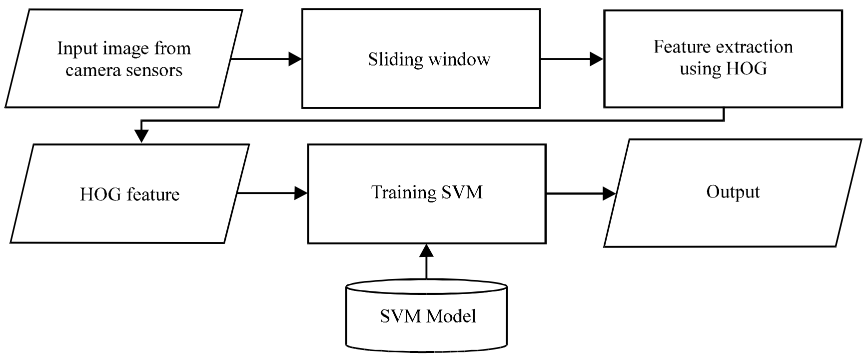

Fig. 5: SVM testing process

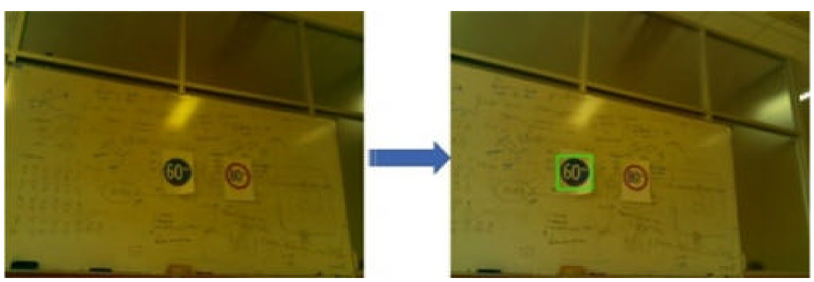

Fig. 6: Sign detection process

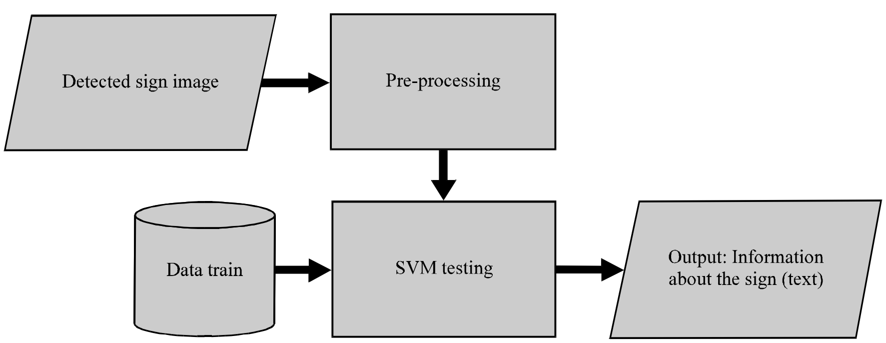

Fig. 7: Sign recognition process

Machine). SVM testing requires some trainer data to train SVM in order to classify new traffic sign image. The used data train is the image signs with a size of 90×90 pixels with JPG format. Once the image is tested with a trained SVM, it will be classified and the system will display its result.

(a) (b)

Fig. 8: a) Sign replica installation and b) Camera placement

RESULTS AND DISCUSSION

The experiments took several parameters to prove the system reliability and require some supporting devices to be like original condition on the highway. The construction of traffic sign followed the regulation of the Minister of Transportation of the Republic of Indonesia in 2014 about traffic signs. Figure 8 shown the installation of traffic sign, followed by the placement of Raspberry Pi camera. Once the program is implemented in Raspberry Pi, we measure the performance of Raspberry Pi in processing the program. At this stage, system testing has the following objectives:

- The detection and recognition of traffic signs
- The parameters to be used
- The system response time

There are 4 parameters to be tested on this experiment and each of the parameters have several different conditions. The parameters used to be analyzed are as follows:

- The number of used training data
- The distance of traffic signs with camera
- Light condition

The experiment is performed 15 times on each parameter then its accuracy rate can be obtained by the Eq. 1:

$$\text{Accuracy rate} = \frac{\text{No.of corectly detected sign}}{15} \times 100\% \quad (1)$$

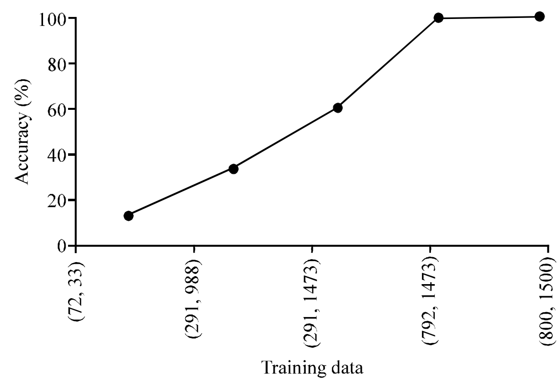

Fig. 9: Result of detection accuracy in various training data: detection accuracy vs camera distance

Detection based on training data: Figure 9 shows the effect of train data on the detection accuracy of traffic sign. There is an increase of detection accuracy after the addition of training data. During detection process there are another objects than the detected sign, therefore, each trial is rated to facilitate the calculation of the detection rate. Giving 1 value, if the number of detected objects equal to 1 and 0 when the system does not detect traffic signs and failed to detects traffic signs. Figure 10a shows the result of detection when the amount of training data using 72 images for positive data and 33 images for negative. It can be seen that the system successfully detects a true traffic sign object but there are still many non-sign object but detected as a sign by the system (false positive).

Meanwhile Fig. 10b shows that the system successfully detects traffic signs appropriately after adding the amount of training data. The increasing in

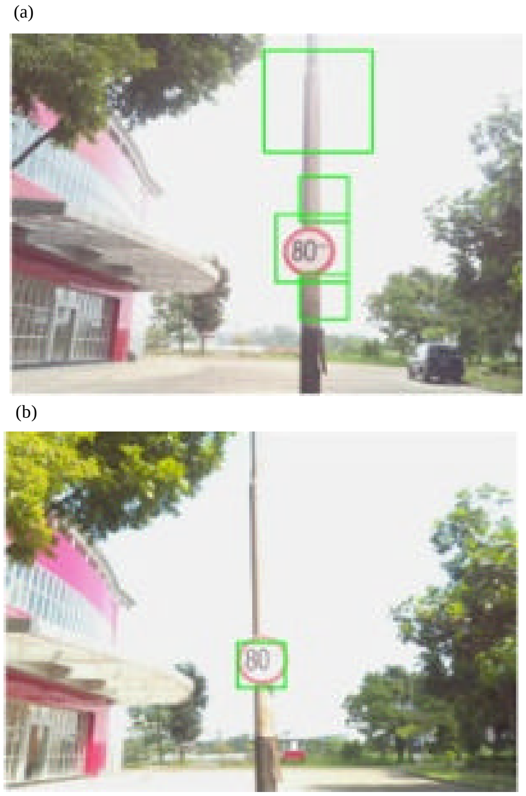

Fig. 10: a, b) Different result of sign detection based on the number of training data

accuracy in line with the addition of training data caused by the number of support vectors which are also increasing. In SVM, determination of the dividing line between two classes or the best hyperlane obtained by measuring the hyperlane margin and searching for its maximum point. Margin is the distance between the hyperplane premises with the nearest pattern of each class. The closest pattern is called a support vector. Table 1 shows the details of support vector number generated from various amount of training data. It can be seen that each addition of training data, the number of support vector is also increasing. With the increase in the number of support vectors, SVM can determine hyperlane with maximum accuracy.

Detection based on camera distance: Figure 11 shows the effect of camera distance on detection accuracy. It can be seen that the highest detection rate is obtained at a distance of 2-6 m when a distance of 8-12 m caused gradual degradation in accuracy until the system is unable to detect the traffic sign at all. Inside the detection system, the sliding window size used is 60×60 pixels and the multiscale multiplier value is 1.5. It can be analyzed that to get the maximum accuracy, the minimum size of the

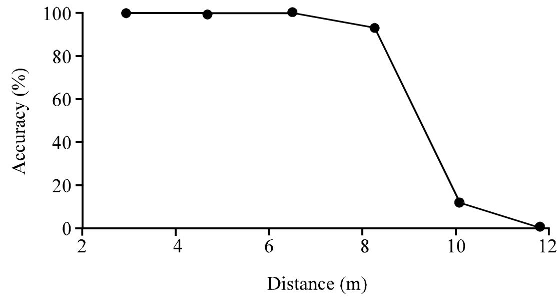

Fig. 11: Result of detection accuracy in various camera distace: detection accuracy vs camera distance

Table 1: Number of support vector in various number of training data

No. of training data		No. of support vector		
Negative data	Positive data	Negative data	Positive data	Total
72	33	7	14	21
291	988	195	61	256
291	1473	298	55	353
792	1473	287	166	453
800	1500	297	163	460

Table 2: Accuracy of detection in various light condition

	Light condition (lux)	
Variables	3441-6513	5072-7271
No. of correctly detected sign	9	15
Accuracy rate (%)	60	10

captured image should be the same size as the sliding windows which is 60×60 pixels. From the experiment, if the camera distance is more than 6 m, the image size of traffic sign get smaller than the sliding window and cause a decrease of detection accuracy. Eventually, at a distance of 12 m the system is unable to detect traffic signs.

Detection based on light condition: Table 2 shows the detail comparison of detection accuracy levels in various light condition. Based on the scenario, there are two light conditions tested and the result has a substantial difference in detection accuracy value. The light value of 5072-7221 lux (afternoon) has a higher detection rate than the light value of 3441-6513 lux (morning) because in the morning condition, the sunlight leads directly to the surface of the sign, causing the surface of the traffic sign object to get full light from the light source. This condition is called front light as shown in Fig. 12 a, therefore, due to the limitations of the camera module that is being used the traffic sign object cannot be detected by the system. Meanwhile, Fig. 12b shows that camera can detect the traffic sign precisely in the afternoon light condition.

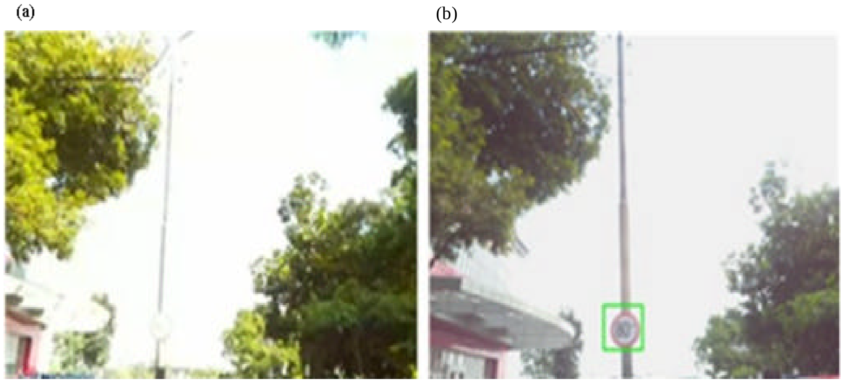

Fig. 12 a, b): Different result of sign detection based on the light condition

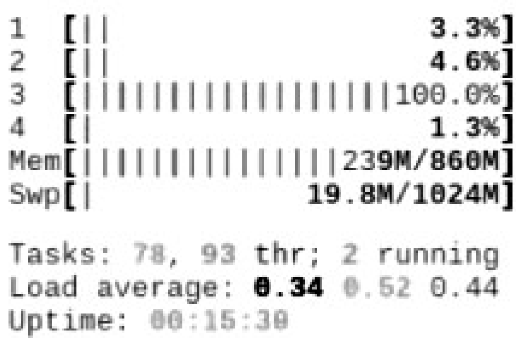

Fig. 13: Performance of Raspberry Pi during detection process

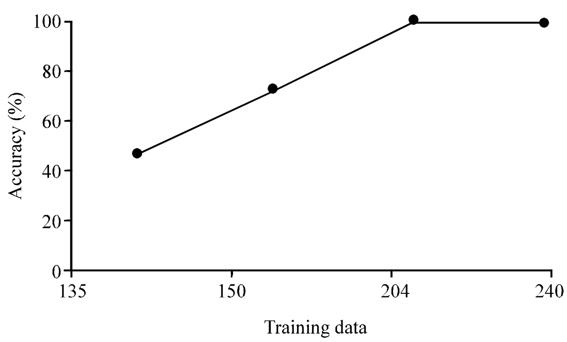

Fig. 14: Result of recognition accuracy in various training data

Hardware performance on detection: Figure 13 shows performance of Raspberry Pi during detection process. It can be seen that only one core that works on maximum load due to the limitation of program.

Recognition based on training data: Figure 14 shows the effect of train data on the recognition accuracy of traffic sign. There is an increase of recognition accuracy after the addition of training data. When the amount of training data above 204 images, the value of the recognition rate is stable at 100% and when the training data added up to 240 images, the accuracy is still 100%. It is same like the previous experiment, the increasing in accuracy in line with the addition of training data caused by the number of support vectors which are also increasing. Table 3 shows the details of support vector number generated from various amount of training data in different vehicle speed. It can be seen that each addition of training data, the number of support vector is also increasing. Table 3 and 4 shows the result of processing time in recognizing the traffic sign. The average time in the recognition process reached 15.93 seconds. It is caused by the performance limitation of Raspberry Pi. When the program runs on Raspberry Pi 3B, only one core that works on maximum load as shown in Fig. 15.

Table 3: Number of support vector in various number of training data in different vehicle speed

	No. of support vector			
No. of training data	20 km/h	40 km/h	80 km/h	Total
135	19	19	22	60
150	20	20	24	64
204	25	26	22	73
240	31	31	19	82

Table 4: Processing time in recognizing the traffic sign

Road sign (km/h)	Recognition result (km/h)	Explanation	Time (sec)
80	80	Recognized	17
80	80	Recognized	15
80	80	Recognized	16
80	80	Recognized	16
80	80	Recognized	16
20	20	Recognized	16
20	20	Recognized	17
20	20	Recognized	15
20	20	Recognized	15
20	20	Recognized	17
40	40	Recognized	15
40	40	Recognized	16
40	40	Recognized	17
40	40	Recognized	14
40	40	Recognized	17
	Average		15.93

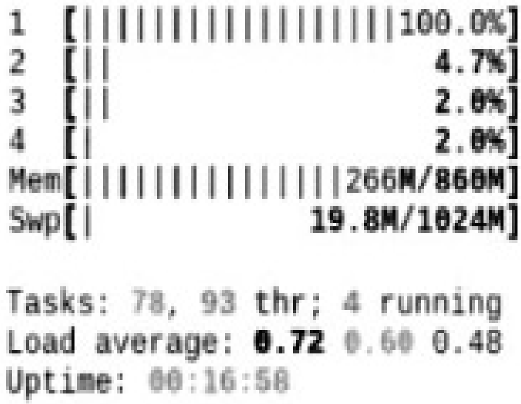

Fig. 15: Performance of raspberry Pi during recognition process

CONCLUSION

This research has shown that a traffic sign detection and recognition system with feature extraction of histogram of oriented gradient and SVM classifier can be done in daylight conditions around 5072-7271 lux with the distance between traffic signs and camera as far as 2-6 m. Test results show the system which is used Raspberry Pi needs average time 15.93 sec for recognizing the detected road sign and yields accuracy more than 80% in both detection and recogniton process. However, this result can still be improved, especially, the processing time, since, the Raspberry Pi only utilizes one of the four cores it has.

REFERENCES

Ardianto, S., C.J. Chen and H.M. Hang, 2017. Real-time traffic sign recognition using color segmentation and SVM. Proceedings of the 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), May 22-24, 2017, IEEE, Poznan, Poland, ISBN:978-1-5090-6345-1, pp: 1-5.

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.

Ellahyani, A. and M. El Ansari, 2016. Complementary features for traffic sign detection and recognition. Proceedings of the 2016 IEEE/ACS 13th International Conference on Computer Systems and Applications (AICCSA), November 29-December 2, 2016, IEEE, Agadir, Morocco, ISBN:978-1-5090-4321-7, pp: 1-6.

Kim, S. and S. Kwon, 2015. Improvement of traffic sign recognition by accurate ROI refinement. Proceedings of the 2015 15th International Conference on Control, Automation and Systems (ICCAS), October 13-16, 2015, IEEE, Busan, South Korea, pp: 926-928.

Santosa, B., 2010. Tutorial support vector machine. Support Vector Mach., 1: 1-23.

Sugiharto, A. and A. Harjoko, 2016. Traffic sign detection based on HOG and PHOG using binary SVM and k-NN. Proceedings of the 2016 3rd International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE), October 19-20, 2016, IEEE, Semarang, Indonesia, ISBN:978-1-5090-1434-7, pp: 317-321.

Zhang, M., H. Liang, Z. Wang and J. Yang, 2014. Real-time traffic sign detection and recognition for intelligent vehicle. Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, August 3-6, 2014, IEEE, Tianjin, China, ISBN:978-1-4799-3978-7, pp: 1125-1131.