Evolution in Electrical and Electronic Engineering Vol. 2 No. 1 (2021) 177-185
© Universiti Tun Hussein Onn Malaysia Publisher's Office

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Speed Estimation of Moving Vehicle based on Input Images from a Smartphone Camera

Nuramin Fitri Aminuddin[1]*, Zarina Tukiran[1,2,4]*, Ariffuddin Joret[1,2], Herdawatie Abdul Kadir[1,3], Shaharil Mohd. Shah[5]

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

[2]Internet of Things (IoT) Focus Group,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

[3]Group of Robotic Engineering and Technology (GREaT) Focus Group,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

[4]Microelectronic and Nanotechnology Shamsuddin Research Center (MINT-SRC),
Institut Kejuruteraan Integrasi (I2E), Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

[5]Advanced Telecommunication Research Center (ATRC),
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2021.02.01.020
Received 03 April 2021; Accepted 08 April 2021; Available online 30 April 2021

Abstract: Using a mobile smartphone camera, this work explored an alternative approach for estimating the speed of moving vehicles. A hardware prototype is developed to monitor the linear movement of vehicle, then the position of the vehicle is detected by using background subtraction as well as tracked using Gaussian Mixture Models (GMM). By comparing the pixel velocity and the measured speed of vehicle, the average pixel velocity for specific speed is determined. The initial findings show promising results to utilise smartphone technology for traffic monitoring application.

Keywords: Smartphone, Camera, Speed Estimation, Moving Vehicle, Image

1. Introduction

Traffic harm is the foremost threat facing the city of Batu Pahat, Johor as a metropolitan trade centre in Malaysia. One solution to minimising the amount of harm is to increase traffic management, which monitors the volume of vehicles and restricts their speed on the road. Usually, traffic management involves predicting the effect of applying diverse regulations of traffic management through performing

**Corresponding author: zarin@uthm.edu.my*

traffic modelling [1]. Traffic modelling determines to assess the results of traffic management measures before any implementation by evaluating traffic data collected from traffic sensors [1]. There are several traffic sensors used in recent years. These sensors are split into two groups depending on their methods: either intrusive or non-intrusive [2]. Intrusive sensors, typically based on magnetic loop detectors, are commonly used but have complex installation and handling, and can be weakened by wear and tear. According to [2], non-intrusive sensors, such as laser markers and Doppler radars, have many drawbacks, such as double-bounce and shading defects, which arise in various vehicles with distinguishable heights. It is also impractical to mount these sensors at traffic sites, as these sensors need regular maintenance [3].

In the latest optical camera and digital imaging technology, an analysis of the traffic image can be rendered to determine the count and speed of the moving vehicles [4]. Therefore, any readily available surveillance camera can be used as an alternative traffic sensor for traffic modelling. The aim of this proposed work is to use a mobile smartphone camera as a surveillance camera to count moving vehicles along a lane and measure their speed in real-time for traffic modelling. A hardware prototype is developed to monitor the linear movement of vehicle and the position of the vehicle is detected by using background subtraction technique as well as tracked using Gaussian Mixture Models (GMM) to determine the speed of the moving vehicle. This paper focused the discussion on technique used to determine moving vehicle speed based on images taken by mobile smartphone camera.

2. Materials and Methods

The proposed system framework is based on the relative positioning scenario in the real world with a moving vehicle as illustrated in Figure 1.

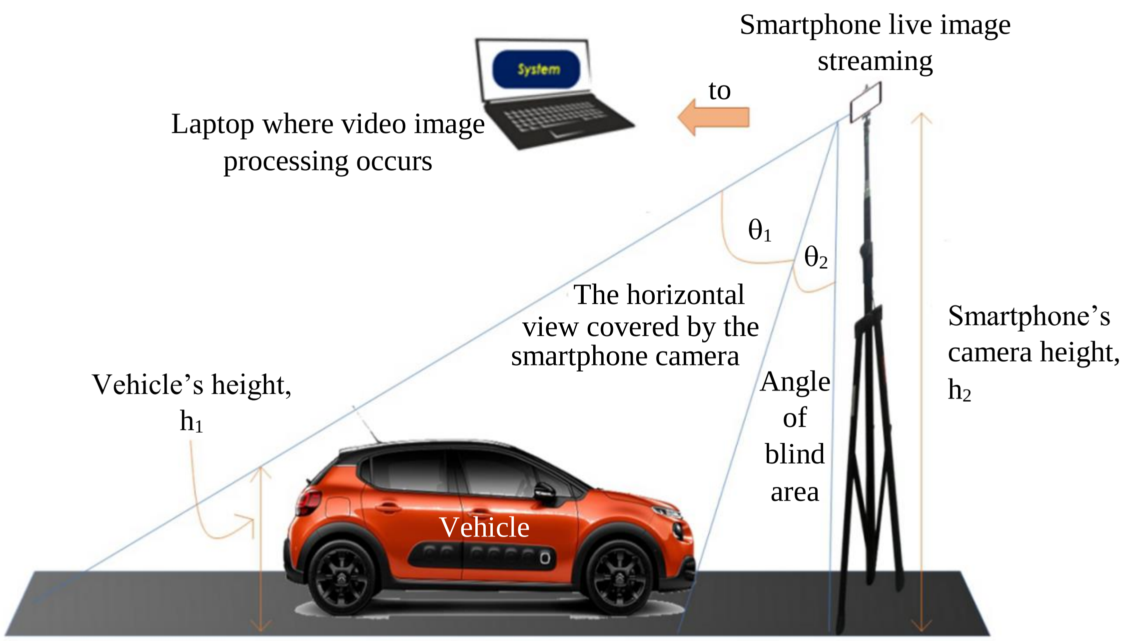
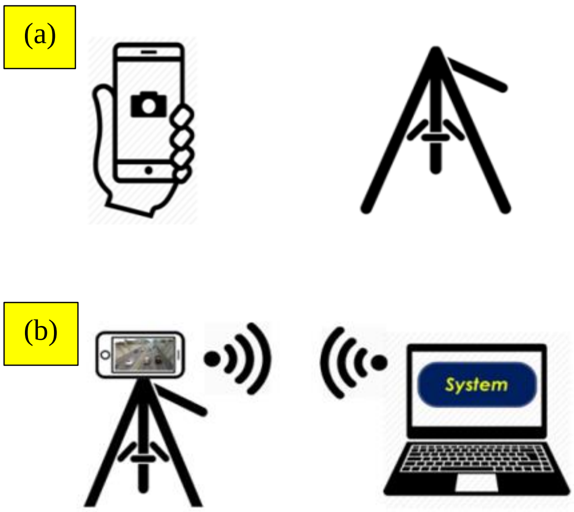


Figure 1: Framework for calibrating the test vehicles

The traffic images are taken from a mobile smartphone camera, OPPO A37 that is mounted on a portable steel tripod as shown in hardware setup of Figure 2. The camera has an overall resolution of 960 x 720 pixels with a focal length of 31mm, a horizontal dimension of 24mm with an image format of 35mm and a maximum resolution of 30fps. The resolution and rate of fps are selected to provide sufficient detail in the image to identify individual vehicles and to capture sequential images rapidly enough so that individual vehicles can be tracked between images without shifting more than one vehicle length between images. The images captured are then transmitted to a laptop, which acts as system processing device with an Intel Core™ i3 processor 1.9 GHz PC with 4GB RAM running on Windows 10, as shown in Figure 3. Meanwhile, the software used is MATLAB version 9.0, R2016a which is the fully integrated editor, compiler and debugger for the system.

Figure 2: Portable steel tripod



Placement of mobile device on a portable tripod. A smartphone is a mobile device used as a temporary electronic road traffic live recording design. The device is deployed in near proximity to the roadway.

Enable a Wi-Fi setting on the smartphone to provide wireless communication to a targeted mobile device. In this work, the targeted mobile device is a laptop, used as a processing device.

The smartphone streams the captured video to the processing device automatically via Wi-Fi connection. The system executed in MATLAB software; estimates speed of vehicular traffic passing along the roadway.

Figure 3: Operation of the hardware

The smartphone is fixed in a horizontal position for the CVF (Camera's View Field) angle. The camera streams the real-time traffic images to the laptop for video image processing processes in MATLAB. The processes start with real-time segmentation of moving regions in image frames, which is an essential step in vehicle detection. One of renown methods for detecting moving objects is background subtraction. One of the successful background subtraction methods is background mixture model [5], which is a Gaussian mixture-based background model (GMM). It proposed a method to construct background pixel model by a mixture of K Gaussian distributions (K = 3 to 5). By analysing the weights of the mixture, which represent the time of colours in the scene, the background can be obtained, which colours stay longer and more static.

The GMM was implemented in the background subtraction as it can handle multi-modal settings such as sky and trees likewise a sturdier detection. Every value of the pixel is modelled by GMM. The Gaussian's mixture variance is determined through which Gaussians reciprocate to the colour of background. Values of pixel that does not settle for background distributions will be considered as foreground. Recursive equations are used to constantly update the parameters of a GMM [6]. This method gives better adaptability to facing illumination changes in the scenes. To detect the vehicle, adaptive background mixture model [7] is used by finding foreground mask in each frame. The foreground mask is obtained from the current frame minus the background model, then the different result is filtered by using a threshold. For moving objects, the obtained contour is combined with its convex hull. Finally, vehicle detection is performed at the midpoint of the bounding rectangle of convex hull for each frame in green colour as depicted in Figure 4.

Figure 4: Moving vehicle detection result

After the vehicle was detected by system, its movements in binary blobs were analysed and tracked using the blob analysis. If the path trans-versed of binary blobs within a certain period of observation are sufficiently long, then the tracked blobs are considered as a moving vehicle. This is a motion-based approach which the system differentiates the motion of moving vehicle from the still motion of background image. This approach is not dependent on vehicle visual features as it provides a shorter computational time that makes it appropriate for real-time systems [8]. By using blob matching operation, the system would able to track binary blobs over an arbitrarily long image sequence. The recommended way is to track every blob in between returning output parameter of sequences of image frames in the bounding box produced, when the vehicle was detected.

In this work, the measurement of vehicle speed is determined by analysing the traces of an object movement in image projection. The analysis is divided into two parts. The first part can be done by calculating the actual movement distance through the vehicle's velocity. The second part is the change of coordinates between the world coordinate system and the coordinate system in the image obtained through the *distance-to-pixel* mapping table. The mapping table is obtained by calibrating the mobile smartphone camera.

2.1 Vehicle Velocity

To simplify the task, speed velocity of moving vehicle is represented by pixel velocity where matching pixel mapping is applied to compensate real speed of moving vehicle to the image domain. Velocity can be defined as the rate of change of position with respect to time.

$$V = \frac{dx}{dt} \tag{1}$$

where *V* is the velocity, *x* is the displacement vector and *t* is the time taken in second. Displacement vector can be determined by calculating the distance between the vehicle current location and its previous location. Since the tracking is only in 2-D (x,y) and only Euclidean distance is taken into account, the simplest way to determine the displacement is by using Pythagoras theorem [9] which is shown in Figure 5.

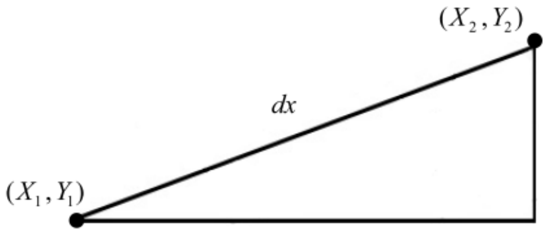


Figure 5: Pythagoras theorem

Displacement can be obtained at,

$$dx = \sqrt{(X_2 - X_1) + (Y_2 - Y_1)} \tag{2}$$

which, the location of centroid is being showed in current frame of *n* and previous frame of *n-1* for a single vehicle, using the (X_2, Y_2) coordinate and (X_1, Y_1) coordinate.

$$n = (X_2, Y_2) \text{ and } n\text{-}1 = (X_1, Y_1) \tag{3}$$

When the displacement is obtained, the unit will be in pixels. The detection system was designed to capture the displacement of moving vehicle in every 1 second, where *dt* is the time. This resulted the vehicle velocity in pixels per second unit. The unit was displayed in four decimal places as designated. For the speed estimation of vehicle, only the magnitude of vehicle velocity is used, whereas *km/h* (kilometre per hour) been used for speed standardization.

2.2 Camera Calibration

To obtain pixels per meter, there is a need to find perpendicular CVF, *P* in camera positioning. By comparing *P* to pixel width of image frame which is 960 pixels, thus pixel per meter can be obtained. Figure 6 shows the camera calibration in term of positioning and labelling of different parameters used for calculation of *P* [10], where *P* is perpendicular CVF divide by actual image width, θ_1 is the horizontal angle of view covered by the smartphone camera, θ_2 is the blind angle, θ_3 is the angular CVF, h_1 is vehicle height, h_2 is smartphone camera's height, d_1 is real distance between smartphone camera and vehicle, d_2 is the horizontal distance between smartphone camera and vehicle and d_3 is blind area.

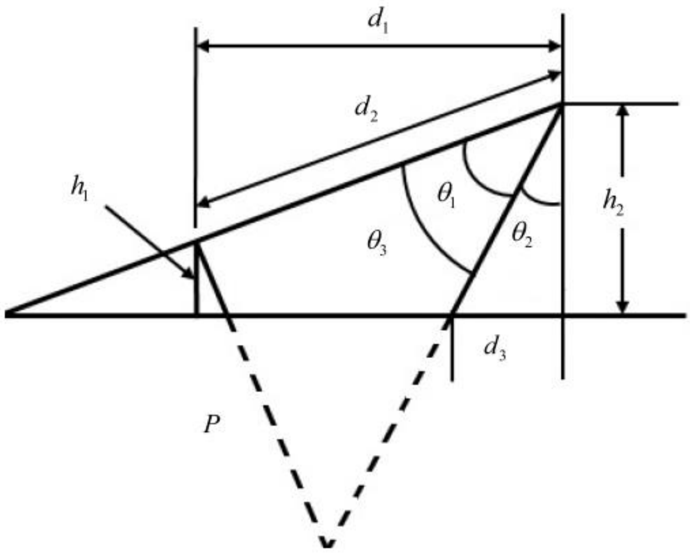


Figure 6: Positioning of camera

From Figure 6, it can be deduced to

$$P = 2d_2 \tan\left(\frac{\theta_1}{2}\right) \tag{4}$$

$$d_2 = \sqrt{(h_2 - h_1)^2 + (d_1)^2} \tag{5}$$

substitute (5) in (4) yield

$$P = 2\sqrt{(h_2 - h_1)^2 + (d_1)^2} \tan\left(\frac{\theta_1}{2}\right) \tag{6}$$

when $\theta_3 \rightarrow 90^o$ then $d_2 \rightarrow d_1$, hence the eq. (6) can be simplified as

$$P = 2d_1 \tan\left(\frac{\theta_1}{2}\right) \tag{7}$$

To find θ_1 in (7),

$$\theta_1 = 2 \arctan\left(\frac{v}{2f}\right) \tag{8}$$

where v is vertical dimension of smartphone camera image format and f is focal length of smartphone camera. The conversion from meter per second to *kph* can be obtained by,

$$kph = \left(\frac{m}{s}\right) 60s60m\left(\frac{1}{1000}\right)$$

$$kph = \left(\frac{m}{s}\right) 3.6 \tag{10}$$

3. Results and Discussion

In order to obtain pixel velocity, a speed range needs to be fixed. The speed of vehicle is measured in the unit of *kph* by referring to the speedometer of test vehicle. A simple experiment is executed to determine the best speed range. The test lane is setup and the distance to travel is predetermined from 15 metres to 45 metres. As higher camera fps will decreases the computational time of detection process [11], therefore the camera maximum limit of 30 fps is proposed as OPPO A37 smartphone is used in this work. Based on the finding in Table 1, the speed range of 10 kph to 40 kph is selected with maximum distance to travel of 35 meters.

Table 1: Distance and Speed of four moving vehicles using smartphone camera

Smartphone camera properties	Distance, m	Measured Speed , kph
31mm focal length and 30 fps	15	18
	25	29
	35	39
	45	47

To find *P* or actual image width in equation (7), the first to obtain is θ_1 in equation (8) which is 42.3225º. The value is approximated to 43º for easier view angle positioning. The distance of d_2 in equation (5) is 35 meters. The distance of d_1 in equation (7) is 32.9018 meters. Hence, *P* is 25.4723 meter while h_1 is approximately 1 meter and h_2 is approximately 3 meter above the ground. Comparing *P* to pixel width of image frame which is 960 pixels, thus 1 pixel is equivalents to 0.0270 meter. As tabulated in the Table 2, the average pixel velocity is then computed from the collected pixels velocity in the duration of 1 second for ten times the test vehicle passing the distance at test lane area. From Table 2, it is observed that the relation between the speed and the pixel velocity is not strictly proportional as at certain testing where the test vehicle is travelling, the number of pixels decreased and vice-versa. The reason might be attributed to the shadow points of vehicle. The recorded pixels still considered acceptable if it is within the range of ±20 pixels for each test.

Table 2: Pixel velocity for the measured speed

	Test#	Measured Speed by Speedometer			
		10 kph	20 kph	30 kph	40 kph
Pixel velocity	1	91.4356	192.8943	305.4476	420.3266
	2	89.4577	199.5675	310.4389	415.6983
	3	95.6784	199.9934	295.6729	420.7733
	4	97.9984	190.3478	300.7771	423.6582
	5	98.7675	189.2213	315.6435	412.6748
	6	92.3452	197.4567	314.5973	420.7766
	7	96.4537	189.4582	302.4434	419.2548
	8	102.4574	192.6759	295.4755	423.8946
	9	95.8972	192.7459	300.8991	420.7689
	10	97.8659	184.5632	300.7143	421.5422
Average Pixel Velocity		95.8357	192.8924	304.2110	418.9368

The average pixel velocity is then mapped to the respective speed which is from 10 kph to 40 kph in the proposed system. The proposed system is developed using MATLAB software and installed in the laptop. The proposed system is then tested by streaming the video from the mobile smartphone camera in real-time. The video contains of vehicle travelling on the test lane with specific speed measured by vehicle's speedometer for four times. The measured speedometer is recorded manually and then compared with the value estimated by the proposed system. Table 3 shows the summary of the findings. Whilst, Figure 7 shows an example of the accomplished result by the proposed system. On the second interface of Figure 7, on top left corner, the first and second yellow box show number of moving vehicle and its speed detected by the proposed system, respectively.

Table 3: Comparison of estimated speed and measured speed

	Test#	Measured Speed by Speedometer in *kph*			
		10	20	30	40
Estimated speed by the proposed system in *kph* (using pixel mapping technique)	1	10	20	30	40
	2	10	20	30	40
	3	10	20	30	40
	4	10	20	30	40

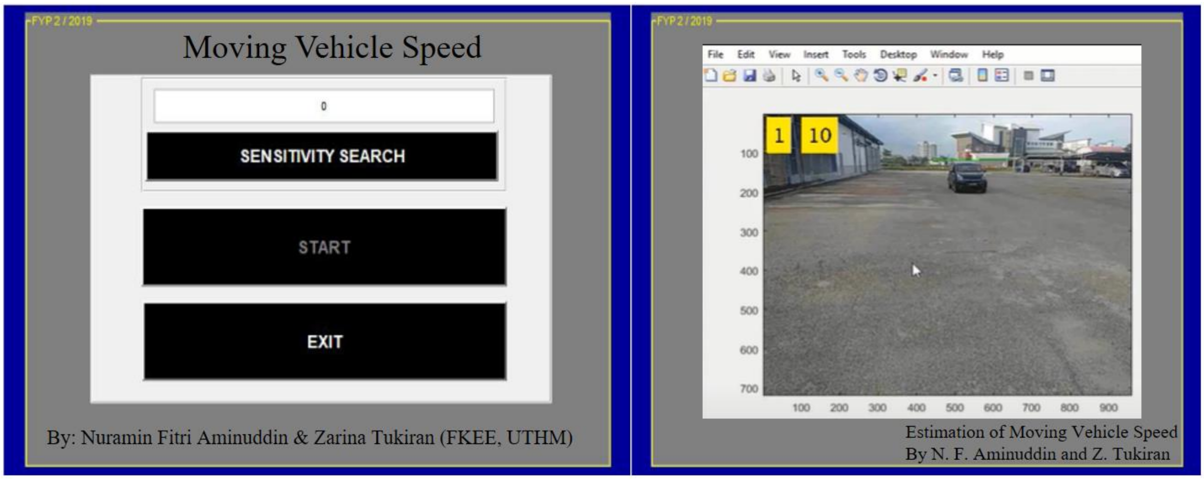


Figure 7: GUI of the proposed system

4. Conclusion

This paper discussed the speed estimation technique by using input image from the mobile smartphone camera. Initial findings show promising results to utilise smartphone technology for traffic monitoring application. This work will be continued to perform a series of field test to evaluate the performance of the proposed system.

Acknowledgement

This research was made possible by Collaborative Research in Engineering, Science and Technology Centre (CREST) Institute. The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] S. e Luca, R. Di Pace, S. Memoli, and L. Pariota., "Sustainable Traffic Management in an Urban Area: An Integrated Framework for Real-Time Traffic Control and Route Guidance Design," in *Sustainability*, vol. 12, no. 2, pp. 726, Jan 2020, doi: 10.3390/su12020726

[2] K. A. Kumar, "Digital Vehicle Speedometer and Setting a Password for Limiting a Vehicle Speed, " in *Int. Res. J. Eng. Tech.*, vol. 5, no. 3, pp. 2936–2938, Mar. 2018, e-ISSN 2395-0056

[3] A. Abdelhalim and M. Abbas, "Towards Real-time Traffic Movement Count and Trajectory Reconstruction Using Virtual Traffic Lanes," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 2527-2533, doi: 10.1109/CVPRW50498.2020.00304

[4] E. Vakili, M. Shoaran, and M.R. Sarmadi, "Single–camera vehicle speed measurement using the geometry of the imaging system," in *Multimed Tools Appl*, **79, pp.** 19307–19327, Mar 2020, doi: 10.1007/s11042-020-08761-5

[5] Y. Liu, Z. Cai and X. Suo, "A Multiframes Integration Object Detection Algorithm Based on Time-Domain and Space-Domain," in *Mathematical Problems in Engineering*, vol. 2018, pp. 1-15, Feb. 2018, doi: 10.1155/2018/4127305

[6] Y. Zhao, A. K. Shrivastava and K. L. Tsui, "Regularized Gaussian Mixture Model for High-Dimensional Clustering," in *IEEE Transactions on Cybernetics*, vol. 49, no. 10, pp. 3677-3688, Oct. 2019, doi: 10.1109/TCYB.2018.2846404

[7] A. Varghese and G. Sreelekha, "A Robust, Low-Complexity Real-Time Vehicle Counting System for Automated Traffic Surveillance," *2020 National Conference on Communications (NCC)*, Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/NCC48643.2020.9056045.

[8] R. Bhardwaj, A. Dhull, and M. Sharma, "A Computationally Efficient Real-Time Vehicle and Speed Detection System for Video Traffic Surveillance," In: Bansal P., Tushir M., Balas V., Srivastava R. (eds) Proceedings of International Conference on Artificial Intelligence and Applications. Advances in Intelligent Systems and Computing, vol 1164. Springer, Singapore, 2020, pp. 583-594, doi: 10.1007/978-981-15-4992-2_55

[9] F. Valocký, P. Drahoš and O. Haffner, "Measure distance between Camera and Object using Camera Sensor," *2020 Cybernetics & Informatics (K&I)*, Velke Karlovice, Czech Republic, 2020, pp. 1-4, doi: 10.1109/KI48306.2020.9039879

[10] W. Czajewski and M. Iwanowski M, "Vision-Based Vehicle Speed Measurement Method," In: Bolc L., Tadeusiewicz R., Chmielewski L.J., Wojciechowski K. (eds) Computer Vision and Graphics. ICCVG 2010. Lecture Notes in Computer Science, vol. 6374. Springer, Berlin, Heidelberg, 2010, pp. 308-315, doi: 10.1007/978-3-642-15910-7_35

[11] K. Yabuuchi, M. Hirano, T. Senoo, N. Kishi, and M. Ishikawa, "Real-Time Traffic Light Detection with Frequency Patterns Using a High-Speed Camera," in *Sensors*, vol. 20, no. 14, pp. 4035, July 2020, doi: 10.3390/s20144035