



# Improved vanishing point reference detection to early detect and track distant oncoming vehicles for adaptive traffic light signaling

Yoanda Alim Syahbana<sup>1,\*</sup>, Dodi Zulherman<sup>2</sup>, Yokota Yasunari<sup>3</sup>

<sup>1</sup>Computer Engineering, Information Technology Department, Politeknik Caltex Riau

<sup>2</sup>Graduate School of Engineering, Gifu University, and  
Faculty of Telecommunication and Electrical Engineering, Institut Teknologi Telkom Purwokerto

<sup>3</sup>Department of Electrical, Electronics, and Computer Engineering, Gifu University

<sup>1</sup>Umban Sari, No. 1, Riau 25265, Indonesia

<sup>2,3</sup>Yanagido 1-1, Gifu 501-1193, Japan

<sup>2</sup>Jl. D. I. Panjaitan, No. 128, Purwokerto 53147, Indonesia

\*Corresponding email: [a3921001@edu.gifu-u.ac.id](mailto:a3921001@edu.gifu-u.ac.id)

Received 11 January 2023, Revised 2 April 2023, Accepted 4 April 2023

**Abstract** — Real-time traffic monitoring is essential for operating an adaptive traffic lighting system and plays a significant role in decision-making, mainly signaling in roadworks. When only one lane is accessible due to temporary road blockage, early detection of oncoming vehicles is crucial to minimize bottlenecks near the traffic light that could result in congestion and accidents. This research aimed to enhance the detection and tracking of traffic at a distance from the traffic light. We utilized the vanishing point as a reference for detection and calculated the region of interest. The vanishing point is estimated using weber orientation descriptor (WOD) method and Gabor filtering, while the region of interest is obtained using a combination of background subtraction and frame difference method. In addition, we also used Kalman filtering to track the detected traffic based on the likelihood of each detection to each motion track, and the selected motion track and its detected traffic are categorized as oncoming traffic. We implemented the proposed method on twelve traffic surveillance videos and evaluated the system performance based on how quickly it could detect oncoming traffic compared with the R-CNN method. The proposed method detected target vehicles in an average of 17.75 frames, while the R-CNN method required an average of 63.36 frames. Moreover, the proposed method's precision depends on the number of pixel orientations used to estimate the vanishing point and the definition of the region of interest. Therefore, the proposed method for enhancing the safety and reliability of an adaptive traffic light system is reliable.

**Keywords** – adaptive traffic light, distant oncoming traffic detection and tracking, region of interest, vanishing point

Copyright ©2023 JURNAL INFOTEL

All rights reserved.

## I. INTRODUCTION

Even though roadworks have a negative impact, such as on traffic and the environment [1], it is necessary to maintain a good service for users [2]. Improving road quality by employing continuous roadworks can reduce traffic delays [3]. During roadwork that utilizes one lane for both primary and secondary streams, temporary road blockages in single-carriageways [4] or more use of narrow lanes [5] are commonly employed. Depending on the number of lanes closed by roadworks, the flow on the remaining lanes can be reduced the capacity between 25 and 40 % [2], [6]. In addition, high-speed variation is one of the most

common safety issues in roadworks areas [3], [7], [8]. Heiden *et al.* [9] said a failure to miss roadwork signs could result in various accidents impacting roadwork safety. Therefore, congestion, accident, and diminished capacity are the effects of traffic obstruction in the work zone [5]–[10]. Moreover, as vulnerable users, contractors and plants operating at road construction sites need protection during roadworks operations [6], [11]. Consequently, traffic management is required during the roadwork to protect and control traffic through the roadworks, such as temporary road blockages or speed limits through road works [12].

Temporary road blockages that only allow traffic

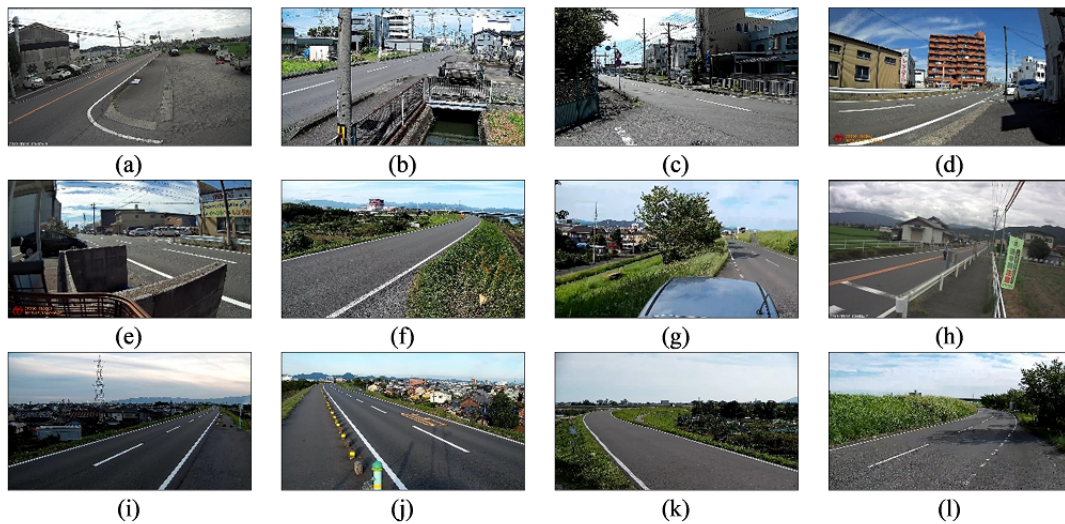


Fig. 1. Sample of a captured scene from video test material: (a) Site 1; (b) Site 2; (c) Site 3; (d) Site 4; (e) Site 5; (f) Site 6; (g) Site 7; (h) Site 8; (i) Site 9; (j) Site 10; (k) Site 11; (l) Site 12.

to flow in one direction at a time can be challenging to manage conventionally by humans as traffic controllers. Due to their vulnerability to accidents, traffic signals are safer for roadworkers as an alternative traffic control method. Even though timing-based signaling systems were commonly used in intersections to reduce congestion [13], timing-based traffic lights could be more effective because they must consider real-time traffic conditions. Instead, it is recommended that adaptive traffic lights be connected to cameras and computers to capture real-time traffic conditions and make decisions about signaling based on the traffic from both sides of the road [14], [15]. These adaptive traffic lights can manage traffic flow more effectively and increase safety.

Adaptive traffic light systems face a significant obstacle in detecting the detection of oncoming traffic, such as the spatiotemporal coverage of traffic information [16]. This early detection provides information for the traffic light system to determine when to signal the appropriate lights, as delayed decisions can lead to sudden stops and potential accidents. However, the camera's perspective projection can make it challenging to detect oncoming traffic, mainly when the vehicles are distant and appear small. The presence of non-vehicle objects can further complicate this situation. Moreover, the low resolution of the vehicle images can limit the effectiveness of artificial intelligence techniques, such as deep learning, for detecting these small objects. Detecting and tracking oncoming traffic precisely is a significant challenge for adaptive traffic light systems [17]–[19].

This study presents a method for detecting oncoming traffic by using the position of the vanishing point in perspective projection as a reference. Weber Orientation Descriptor is a texture-based method to estimate the vanishing point (WOD). Define the

region of interest (RoI) to concentrate detection on the roadway region. This data is used to detect and track oncoming traffic. The proposed method employs background subtraction and Kalman filtering to detect oncoming traffic and track detected objects. Twelve In-Luck Company videos were utilized to test the proposed method. In terms of how early the proposed method could see oncoming traffic, it was compared to the Regions with Convolutional Neural Networks (R-CNN) performance of R-CNN.

This paper's structure is as follows: Section II introduces the video test materials and the design of the proposed method. Then, section III describes the results of the experiment. Section IV provides the evaluation of the proposed method. At the last, section V summarizes the research finding in conclusion.

## II. MATERIALS AND METHOD

This section discusses video test material and our proposed method.

### A. Video Test Material

Video footage for this study was obtained from In-Luck Company, which provides security and traffic guidance services at road construction sites. In-Luck Company positions cameras on the road to record daylight traffic activity. The camera footage shows perspective projection, which makes distant traffic appear small. Fig. 1 shows a sample of the captured scenes from the 12 videos used as test materials. The duration of each video is 10 seconds. The selected duration focuses on one vehicle that is coming toward the camera.

Each video frame is represented as  $I(x, y, t) \in 0, 1, \dots, 255$ , where  $x \in 1, 2, \dots, N_x$ ,  $y \in 1, 2, \dots, N_y$ , and  $t \in 1, 2, \dots, T$ .  $N_x$  and  $N_y$ , are the width and height of the video frame, respectively.

In this study, the video test material has  $N_x = 1920$  and  $N_y = 1080$ .  $T$  is the total video frames that are calculated using (1).

$$T = V_{duration} \times V_{fps} \quad (1)$$

where  $V_{duration}$  (s) and  $V_{fps}$  (frame/s) are the duration of the video and video frame rate, respectively. The video frame rate used is 30 frame/s, except for Site 1 and Site 8, which used 20 frame/s.

$I(x, y, t)$  is a vector that comprises three channels of RGB color expressed in (2).

$$I(x, y, t) = \begin{cases} I_R(x, y, t) \\ I_G(x, y, t) \\ I_B(x, y, t) \end{cases} \quad (2)$$

where  $I_R(x, y, t)$ ,  $I_G(x, y, t)$ , and  $I_B(x, y, t)$ , is a channel of red, green, and blue color, respectively. In this study,  $I(x, y, t)$  is converted to a grayscale image. Grayscale conversion is calculated using (3).

$$I_{x,y,t} = \frac{I_R(x, y, t) + I_G(x, y, t) + I_B(x, y, t)}{3} \quad (3)$$

where  $I_{x,y,t}$  is a grayscale conversion result used as input for the proposed method.

### B. Proposed Method

In this study, we propose a method consisting of three steps: initialization, traffic detection, and tracking oncoming traffic. As the initial step, initialization involves defining the RoI and estimating the vanishing point. The first process involves enclosing the region where the traffic movement occurs in the captured scene. This process employed background subtraction to detect foreground objects and frame difference to detect objects' movement. Then, the objects' movement defines the RoI. The second process aims to estimate the coordinate of the vanishing point from the video frame. This process uses the weber orientation descriptor (WOD) method [20] to calculate the differential excitation of the frame texture features, followed by Gabor filtering to calculate pixel orientation. The information about the differential excitation and orientation from every pixel is then used in a voting scheme to estimate the coordinate of the vanishing point.

The second step uses the RoI and vanishing point in the initial step to detect traffic. It also employed background subtraction to detect objects. The last step tracks the detected traffic by associating it with its movement from frame to frame. We used Kalman filtering to track the detected traffic based on the likelihood of each detection to each motion track. Then, the motion track for traffic moving away from the vanishing point is selected. Finally, the selected motion track and its detected traffic are categorized as oncoming traffic. Fig. 2 shows the design of the proposed method, and the following section provides

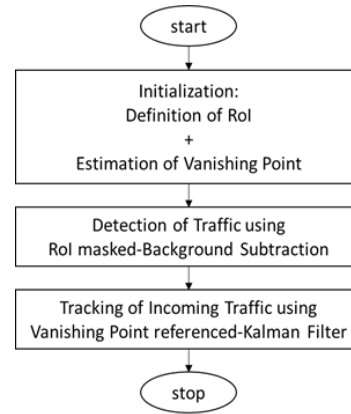


Fig. 2. Design of proposed method.

more details about the steps used in the proposed method.

#### 1) Initialization Step

The initialization step involves two processes as a preparatory action preceding detecting and tracking incoming traffic. The first process in the initialization step is defining the RoI. The sequence of  $I(x, y, t)$ , with the existence of traffic, is used for the RoI definition process. Due to the optimum processing time to obtain the target movement of traffic reason, this study recommends down-sampling  $I(x, y, t)$  for  $t \in V_{fps}, 2V_{fps}, \dots, T/V_{fps}$ . At first, defining the RoI process, the algorithm detects foreground objects using the background subtraction method. The background frame for background subtraction process is calculated using (4).

$$I_{BG}(x, y, t) = \text{med}(I(x, y, t-r), \dots, I(x, y, t), \dots, I(x, y, t+r)) \quad (4)$$

where  $I_{BG}(x, y, t)$  is generated background, and  $r$  is the range of sequential frame calculated using the median. The median calculation is used to minimize brightness fluctuation from sunlight intensity variation. In this step,  $r$  is predefined as 150 frames based on visual observation of detection quality. The predefined value of 150 frames is keep the background updated based on 150 frames (5 seconds for 30 fps and 7.5 seconds for 20 fps video) before and after the current frame condition. In addition, the predefined value is also used to remove any traffic movement that remains static for less than 10 or 15 seconds. Then, the foreground is calculated using (5).

$$I_{FO}(x, y, t) = |I(x, y, t) - I_{BG}(x, y, t)| \quad (5)$$

where  $I_{FO}(x, y, t)$  is detected foreground from background subtraction calculation. The detected foreground still contains non-vehicle objects and noise caused by road texture and shadow variations. Due to noise reduction in the detected foreground, we used a thresholding-based noise removal process.  $I_{FO}(x, y, t)$  values with a lower intensity than the threshold value ( $th$ ) are removed from  $I_{FO}(x, y, t)$ . This study's threshold value selection is predefined for each video

test material. The noise removal in this step is calculated using (6).

$$I_{FO}(x, y, t) = \begin{cases} 1, & I_{FO}(x, y, t) > th \\ 0, & otherwise \end{cases} \quad (6)$$

where  $I_{FO}(x, y, t)$  is redefined after noise filtering to avoid the complexity of notation. In addition, the morphology process, including blurring and dilation, is also applied to remove the noise.

The objective of defining RoI is to minimize the unpredictability of the environment surrounding the roadway. This study focuses on the primary region where vehicle movement occurs. This study uses frame difference to detect significantly moving foreground objects to obtain the RoI. Frame difference is calculated using (7).

$$I_{MFO}(x, y, t) = |I_{FO}(x, y, t) - I_{FO}(x, y, t + 1)| \quad (7)$$

where  $I_{MFO}(x, y, t)$  is detected moving foreground object. Then,  $I_{MFO}(x, y, t)$  from all  $t$  is summed up into one frame that shows the number of regions where moving foreground objects existed. In addition, thresholding is also applied to remove regions with non-vehicle movement. The process is calculated using (8).

$$I_{RoI}(x, y) = \begin{cases} 1, & th_{low} > \sum_{t=1}^T I_{MFO}(x, y, t) \\ & > th_{high} \\ 0, & otherwise \end{cases} \quad (8)$$

where  $I_{RoI}(x, y)$  represent the RoI.  $th_{low}$  and  $th_{high}$ , are minimum and maximum thresholds to filter non-vehicle movement.  $th_{low}$  and  $th_{high}$  are varied for each video material and predefined based on visual observation.

The second process in the initialization step is the estimation of the vanishing point coordinate. The vanishing point in this step is estimated from a single  $I(x, y, t)$  with  $t$  is preselected manually by visual observation. This study recommends selecting  $t$  without the existence of traffic to obtain a precise vanishing point. In this process,  $I_{VP}(x, y)$  is  $I(x, y, t)$  for the preselected  $t$ . In addition,  $I_{VP}(x, y)$  is also convolved with a median filter with a size of  $5 \times 5$  pixels to reduce noise.

The vanishing point estimation is started by calculating two components of WOD: differential excitation and orientation at each pixel location. Differential excitation is calculated based on the difference between center pixel intensity and the average intensity of all neighbors pixel in a  $k \times k$  kernel size. Differential excitation is calculated using (9).

$$\xi_{wod}(p_{center}) = \begin{cases} \sqrt{G(p_{center})}, & G(p_{center}) \geq 0 \\ 0, & otherwise \end{cases} \quad (9)$$

in which,

$$G(p_{center}) = \arctan\left(\frac{p_{center} - \overline{p_{neighbor}}}{p_{center}}\right) \quad (10)$$

where  $\xi_{wod}(p_{center})$ ,  $p_{center}$ ,  $\overline{p_{neighbor}}$ , and  $G(p_{center})$  are differential excitation for  $p_{center}$ , the intensity of the center pixel, the average intensity of all neighbors pixel, and the intensity difference, respectively. This study predefined  $k = 25$  as the size of the kernel.  $\xi_{wod}(p_{center})$  is further processed by thresholding to minimize the noise in the frame texture features. This study defines  $T = 0.05$  as thresholding value. The normalized value of  $\xi_{wod}(p_{center})$  that is larger than  $T$  is used to estimate the vanishing point.

This study uses the Gabor filter to estimate the dominant orientation at each pixel location, as has been used in [21]. Kernel of Gabor filter  $g$  that is centered at  $(x, y)$  for orientation  $\varphi_n$  and radial frequency  $\omega = 2\pi/\lambda$  is defined as seen in (11).

$$g_{\varphi_n}(x, y) = e^{-\frac{1}{8\sigma^2}(4a^2+b^2)} \cdot (ia\omega - e^{c^2/2}) \quad (11)$$

where  $a = x \cos \varphi_n + y \sin \varphi_n$  and  $b = -x \sin \varphi_n + y \cos \varphi_n$ . In this study,  $\sigma = k/9$ ,  $c = 2.2$ , and  $\lambda = k\pi/10$  are a constant, similar to the parameter setting in [20].  $\varphi_n$  is calculated using (12).

$$\varphi_n = \frac{(n-1)\pi}{N_\varphi}, \text{ with } n = 1, 2, \dots, N_\varphi \quad (12)$$

where  $N_\varphi$  is a total number of orientations. Dominant orientation for  $I_{VP}(x, y)$  for each  $p_{center}$  is calculated using (13).

$$\hat{I}_{\varphi_n}(p_{center}) = I_{VP}(p_{center}) * g_{\varphi_n}(p_{center}) \quad (13)$$

where  $\hat{I}_{\varphi_n}(p_{center})$  is the result of convolution between the video frame and kernel of the Gabor filter and  $*$  denotes the convolution operator.  $\hat{I}_{\varphi_n}(p_{center})$  as a convolution result, has a real part and an imaginary part. These two parts are used to calculate Gabor energy for each pixel in  $\hat{I}_{\varphi_n}(p_{center})$ . Gabor energy is calculated using (14).

$$E_{\varphi_n}(p_{center}) = \sqrt{\text{Re}(\hat{I}_{\varphi_n}(p_{center}))^2 + \text{Im}(\hat{I}_{\varphi_n}(p_{center}))^2} \quad (14)$$

where  $E_{\varphi_n}(p_{center})$  is the magnitude of Gabor energy at  $p_{center}$ . Finally, orientation at each pixel location is defined in (15).

$$\theta_{wod}(p_{center}) = \text{Argmax}_{\varphi_n} E_{\varphi_n}(p_{center}) \quad (15)$$

where  $\theta_{wod}(p_{center})$  is  $p_{center}$  orientation.

The vanishing point is estimated based on the line-voting scheme (LVS) result. Firstly, LVS sets accumulator space with the same size as  $I_{VP}(x, y)$  with an initial zero value. Secondly,  $\xi_{wod}(p_{center})$  and its counterpart  $\theta_{wod}(p_{center})$  act as a voter that draws rays in the accumulator space. The corresponding accumulator space is increased by one if the rays lie over it. Finally, the maximum value in the accumulator space is defined as the vanishing point coordinate  $(x_{vp}, y_{vp})$ .

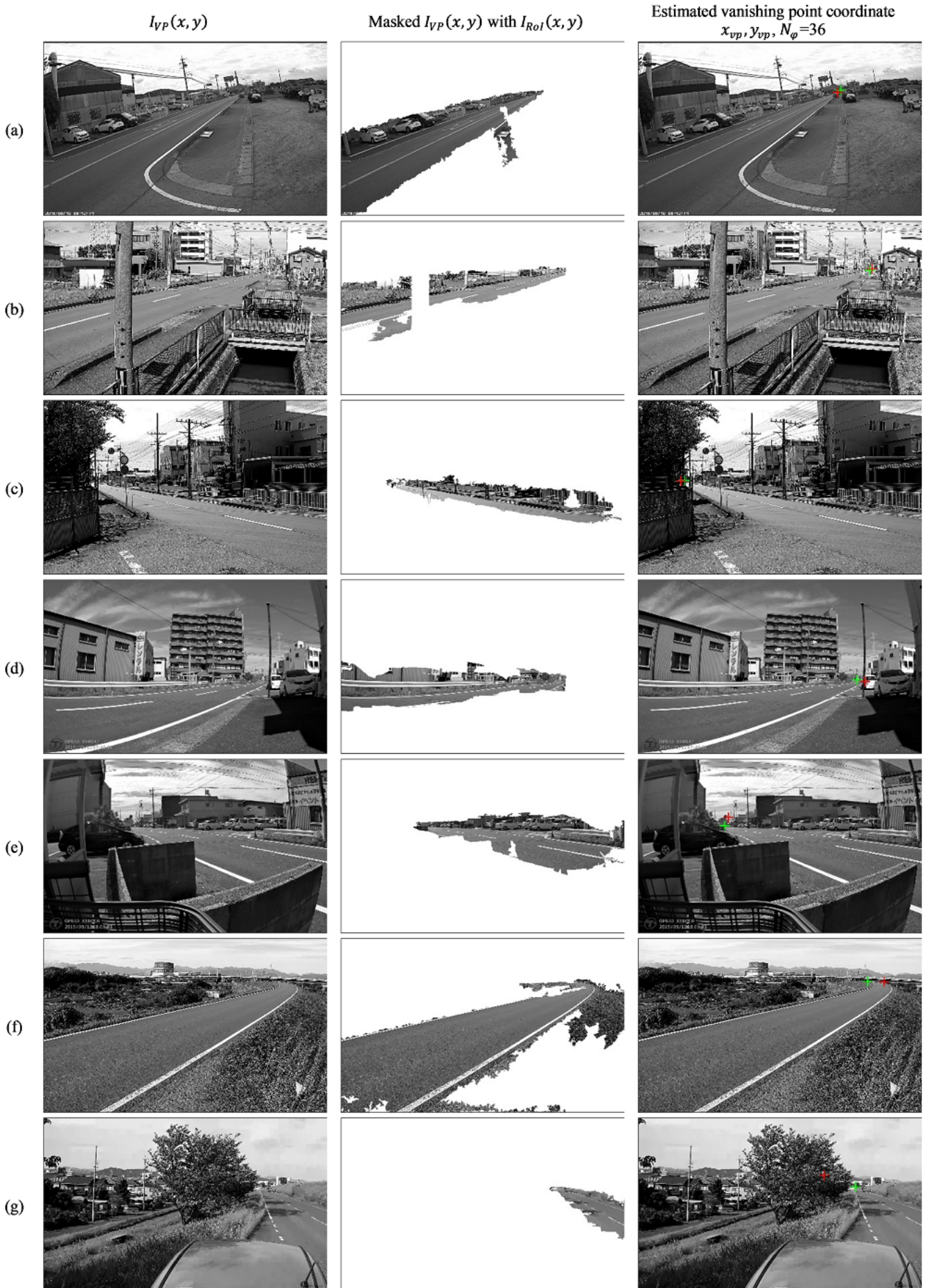


Fig. 3. Result of initialization of the proposed method: (a) Site 1, (b) Site 2, (c) Site 3, (d) Site 4, (e) Site 5, (f) Site 6, (g) Site 7, (h) Site 8, (i) Site 9, (j) Site 10, (k) Site 11, (l) Site 12.

### 2) Object Detection using Background Subtraction

The second step uses the defined RoI from the initialization step to mask the  $I(x, y, t)$ . The masking process conducted in (16).

$$I_{masked}(x, y, t) = \begin{cases} I(x, y, t), & I_{RoI}(x, y) = 1 \\ 0, & otherwise \end{cases} \quad (16)$$

where  $I_{masked}(x, y, t)$  is the masking result. Then, a similar process of background subtraction as used in (4), (5), (6) is applied to all frames in  $I_{masked}(x, y, t)$ .  $I_{FO}(x, y, t)$  is obtained from the process.

### 3) Object Tracking using Kalman Filter

$I_{FO}(x, y, t)$  from the second step is further processed to track its movement. The process is to associate detected  $I_{FO}(x, y, t)$  based on its movement from frame to frame. Since this study uses video from a stationary camera, the Kalman filter [22] can be used to predict object tracks in each frame and determine the likelihood of each detection to each track. Track maintenance is also applied to update the status of any new or disappearing objects in the video frame.

The motion estimation process in this step mainly follows the Matlab documentation [23]. The configuration is set to detect the minimum blob area of 100 pixels to handle this study's low resolution of oncoming traffic. The motion estimation step generates the coordinates of the detected objects in each frame. However, it is possible for the system also to capture outgoing traffic approaching the vanishing point, so the results are filtered to only include detections moving further away from the vanishing point.

The distance of the vanishing point to the detected object is calculated using Euclidean distance. Euclidean distance is calculated in (17).

$$d(t) = \sqrt{(x_{obj}(t) - x_{vp})^2 + (y_{obj}(t) - y_{vp})^2} \quad (17)$$

where  $d(t)$  and  $x_{obj}(t)$ ,  $y_{obj}(t)$  are Euclidean distance of the object and coordinate of the detected object at frame  $t$ , respectively. Filtering oncoming traffic is conducted in (18).

$$incoming(t) = \begin{cases} 1, & d(t) > d(t-1) \\ 0, & otherwise \end{cases} \quad (18)$$

where  $incoming(t)$  is label for oncoming traffic.

## III. RESULT

This section shows the result of the initialization step and the detection and tracking of oncoming traffic.

### A. Initialization Step

Fig. 3 shows the result of the initialization step. The columns represent the result of the grayscale image  $I_{VP}(x, y)$ , the grayscale image masked with defined RoI and overlaid vanishing point, respectively. The

rows represent each video test material. As shown in the second column of Fig. 3, the defined RoI covers the main road lane where most traffic activity occurs. The defined RoI is also affected by perspective projection, which causes one side of the RoI to appear smaller as the road lane gets further away. Although the shape of the RoI may be irregular in some cases, the objective of excluding the movement of non-vehicle objects such as shrubs, grass, trees, and flags is minimized. It can be seen in the second column of Fig. 3: (f-l), where the road is bordered by grass. The third column of Fig. 3 shows the estimated vanishing point from the initialization step as a red crosshair. For comparison, the ground truth of the vanishing point is shown as a green crosshair. In this study, the ground truth of the vanishing point was manually marked using visual fixation.

The qualitative comparison shows that the estimated vanishing point from the initialization step can almost match the ground truth in a straight road, such as in Fig. 3: (a-e). It is because the LVS mainly defines the vanishing point based on the significant orientation of straight edge object that exists in the  $I_{VP}(x, y)$ . The existence of road lines, road fences, pavement, and aerial utility cables influences the accuracy of the estimated vanishing point. In the case of Fig. 3: (f-k), the curved shape of the roadway shifts the estimated vanishing point from the ground truth. In this condition, the curved edge object cannot get a significant vote in the LVS. Observation of the experiment result also shows that the straight edge of bridge and other visible roadways also shifts the vanishing point in the curved road. In addition, Fig. 3: (l) shows how the vanishing point is estimated in an S-curved road. In this case, the edge of the curved road is not visible. Fig. 4 highlights how these objects (highlighted in red ovals) influence the voting process of rays in accumulator space.

As a quantitative comparison, the estimation error of the vanishing point is calculated using normalized Euclidean distance [24]. The estimation error is defined in (19).

$$\delta = \frac{\sqrt{(x_{vp} - x_{gt})^2 + (y_{vp} - y_{gt})^2}}{\sqrt{N_x^2 + V_y^2}} \quad (19)$$

where  $\delta$  and  $(x_{vp}, y_{vp})$  are estimation error and ground truth coordinates, respectively.  $\delta$  near 0 represents a close estimation of the vanishing point to the ground truth; otherwise,  $\delta$  near 1 represents the inaccuracy of estimation.

Table 1 tabulates  $\delta$  for each video test material with a variation of  $N_\varphi \in \{9, 12, 18, 36, 180\}$ . Variation of  $N_\varphi$  influences how accurate the estimation of the vanishing point based on the orientation in the Gabor filter calculation. The estimation error is generally relatively low, with a maximum value of 0.1635. Increasing  $N_\varphi$  means an increased resolution of  $\varphi_n$  precision.

Table 1. Quantitative Comparison of Vanishing Point Estimation Error ( $\delta$ ) for Variation of  $N_\varphi$

Site	$\delta$					$\bar{\delta}$
	9	12	18	36	180	
1	0.0134	0.0204	0.0051	0.0040	0.0080	0.0111
2	0.0947	0.0501	0.0053	0.0027	0.0171	0.0369
3	0.1026	0.0161	0.0025	0.0040	0.0149	0.0288
4	0.0282	0.0188	0.0166	0.0087	0.0059	0.0151
5	0.0538	0.0074	0.0144	0.0099	0.0149	0.0208
6	0.0328	0.0301	0.0428	0.0174	0.0251	0.0312
7	0.0364	0.0163	0.0365	0.0352	0.0115	0.0242
8	0.0221	0.0148	0.0278	0.0154	0.0180	0.0195
9	0.0081	0.0199	0.0222	0.0246	0.0290	0.0205
10	0.1635	0.0278	0.0270	0.0283	0.0398	0.0587
11	0.0390	0.0946	0.0271	0.0375	0.0444	0.0500
12	0.0241	0.0067	0.0724	0.0132	0.0150	0.0259

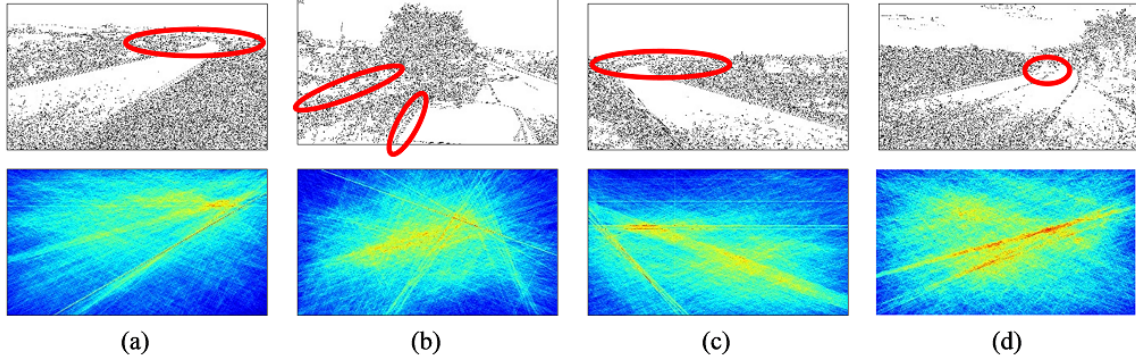


Fig. 4. Sample detected edge (first row) and accumulator space (second row) that influence the estimated vanishing point: (a) Site 6; (b) Site 7; (c) Site 11; (d) Site 12.

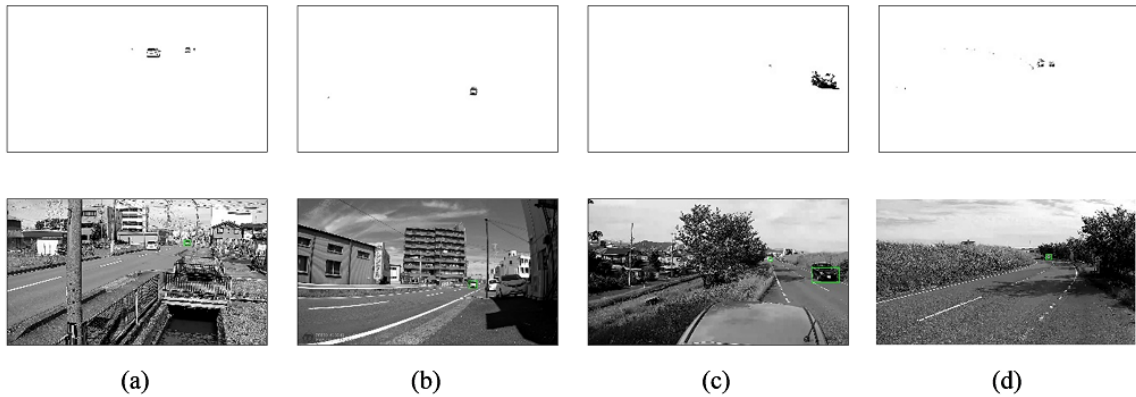


Fig. 5. Sample detection of  $I_{FO}(x, y, t)$  (first row) and final result of oncoming traffic detection (second row): (a) Site 2; (b) Site 4; (c) Site 7; (d) Site 12.

Thus, the error estimation can be lowered. However, the processing time for high-resolution orientation is also increased.

**B. Detection and Tracking of Oncoming Traffic**

Fig. 5 shows a sample result of the traffic detection from the second step. The first row shows the detection of  $I_{FO}(x, y, t)$  inside the RoI. To achieve the study purpose, a small object that appears inside the RoI is categorized as the candidate of the oncoming traffic. Then, the Kalman filter further processes the detection to calculate motion estimation. The second row of Fig. 5 shows the detected object that is getting distant from the vanishing point. These objects are defined as the final result of the proposed method.

**IV. DISCUSSION**

The proposed method's performance is compared to the Regions with Convolutional Neural Networks (R-CNN) [25] object detectors. The evaluation is based on how early the method can detect oncoming traffic.

The R-CNN uses a CNN that consists of an image input layer, 2D convolution layer, rectified linear unit (ReLU) layer, max pooling layer, fully connected layer, softmax layer, and classification output layer for the neural network. The R-CNN uses a CIFAR-10 data set, which contains 50,000 training images used to train a CNN. The training images have ten categories, including automobiles relevant to this study. The training is conducted using stochastic gradient descent with

momentum (SGDM) with an initial learning rate of 0.001. The initial learning rate is reduced every eight epochs for 40 epochs of training.

After confirming that the R-CNN works well with the CIFAR-10 data set, the network is also trained using a self-generated data set. This data set is created based on video frames from the first and second cameras, and each image is labeled manually based on visual observations from the frames. The training set consists of 40 images of vehicles, which represent oncoming traffic and are varied in size, position in the roadway, and grayscale intensity. The training is conducted using the same SGDM algorithm with an initial learning rate of 0.001 for 100 epochs. The implementation of the R-CNN in this study follows the Matlab documentation [26] with some modifications to fit the conditions of the study. The benchmarking process begins by selecting sequential frames as the objects of evaluation. These frames are selected manually from each video test material, starting at  $t_{start}$ .

Table 2. Benchmarking Results in the Detection of Oncoming Traffic

Site	$t_{start}$	$t_{end}$	$t_{detect}$	
			Proposed Method	R-CNN
1	870	1070	880	890
2	360	660	370	437
3	120	420	139	202
4	4200	4500	4211	4276
5	1110	1410	1121	1321
6	600	900	644	672
7	1800	2100	1810	1817
8	1080	1280	1090	1120
9	3780	4080	3792	*
10	1080	1380	1094	1132
11	4500	4800	4538	4545
12	690	990	714	695

\*Until  $t_{end}$ , R-CNN cannot detect the target vehicle.

Table 2 summarizes the detection results for the benchmarking process. Overall, the proposed method has earlier detection of oncoming traffic than the R-CNN method. For this study's twelve video test materials, the proposed method requires an average of 17.75 frames to detect the target vehicle. In contrast, the R-CNN requires an average of 63.36 frames to detect the target vehicle. For Site 9, the R-CNN method cannot detect the target vehicle because it is too small to be detected until  $t_{end}$ . This result shows that the R-CNN requires a larger vehicle image to recognize it as a vehicle. For example, in Site 1, the vehicle is detected if it is at least  $96 \times 200$  pixels in size, and in Site 8, it is detected if it is at least  $141 \times 176$  pixels in size. For Site 12, the R-CNN method has earlier detection than the proposed method because the detected vanishing point from the proposed method is located in the center of the roadway due to its S-curved nature. As shown in Fig. 3: (l), the oncoming car must pass the vanishing point before it can be detected as incoming traffic.

## V. CONCLUSION AND FUTURE WORK

This study has successfully achieved the objective of proposing a method based on a vanishing point reference. The proposed method shows improved performance in detecting and tracking distant oncoming traffic. The result demonstrated that the RoI could be defined, and the vanishing point can be estimated—furthermore, the proposed method results in earlier detection than the R-CNN method. The results also suggest that the proposed method's performance depends on the definition of the RoI and the number of pixel orientations used in calculating the Gabor filter, both of which affect the accuracy of the vanishing point.

In the future, it will be challenging to experiment with the proposed algorithm in different environments, such as Indonesian traffic. The different environments will have different traffic and local conditions that must be considered. In addition, it is also interesting to experiment with more than one incoming/outgoing traffic. Therefore, the proposed algorithm must also be tuned in this new environment.

## ACKNOWLEDGMENT

We send special thanks to In-Luck Company for data collection support.

## REFERENCES

- [1] P. Lepert and F. Brillet, "The overall effects of road works on global warming gas emissions," *Transp. Res. Part D Transp. Environ.*, vol. 14, no. 8, pp. 576–584, 2009, doi: 10.1016/j.trd.2009.08.002.
- [2] S. Yousif, M. Alterawi, and R. R. Henson, "Red light running and close following behaviour at urban shuttle-lane roadworks," *Accid. Anal. Prev.*, vol. 66, pp. 147–157, 2014, doi: 10.1016/j.aap.2014.01.021.
- [3] J. Kim, "Does roadwork improve road speed? Evidence from urban freeways in California," *Reg. Sci. Urban Econ.*, vol. 93, no. July 2021, p. 103773, 2022, doi: 10.1016/j.regsciurbeco.2022.103773.
- [4] N. A. Samoail and S. Yousif, "A study into urban roadworks with shuttle-lane operation," *Math. Transp. Plan. Control*, pp. 79–88, 1998, doi: 10.1108/9780585474182-008.
- [5] S. Yousif, Z. Nassrullah, and S. H. Norgate, "Narrow lanes and their effect on drivers' behaviour at motorway roadworks," *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 47, pp. 86–100, 2017, doi: 10.1016/j.trf.2017.04.016.
- [6] G. Walker and M. Calvert, "Driver behaviour at roadworks," *Appl. Ergon.*, vol. 51, pp. 18–29, 2015, doi: 10.1016/j.apergo.2015.03.019.
- [7] R. T. Steinbakk, P. Ulleberg, F. Sagberg, and K. I. Fostervold, "Speed preferences in work zones: The combined effect of visible roadwork activity, personality traits, attitudes, risk perception and driving style," *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 62, pp. 390–405, 2019, doi: 10.1016/j.trf.2019.01.017.
- [8] J. A. Allpress and L. S. Leland, "Reducing traffic speed within roadwork sites using obtrusive perceptual countermeasures," *Accid. Anal. Prev.*, vol. 42, no. 2, pp. 377–383, 2010, doi: 10.1016/j.aap.2009.08.014.



- [9] R. M. A. van der Heiden, C. P. Janssen, S. F. Donker, and C. L. Merckx, "Visual in-car warnings: How fast do drivers respond?," *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 65, pp. 748–759, 2019, doi: 10.1016/j.trf.2018.02.024.
- [10] C. Schwietering and M. Feldges, "Improving Traffic Flow at Long-term Roadworks," *Transp. Res. Procedia*, vol. 15, pp. 267–282, 2016, doi: 10.1016/j.trpro.2016.06.023.
- [11] Z. Fazekas, G. Balázs, L. Gerencsér, and P. Gáspár, "Locating roadworks sites via detecting change in lateral positions of traffic signs measured relative to the ego-car," *Transp. Res. Procedia*, vol. 27, pp. 341–348, 2017, doi: 10.1016/j.trpro.2017.12.004.
- [12] S. J. Gillard, A. Wilson, J. Slip, P. B. Kenington, and L. Thompson, "Technology for variable speed limits for roadworks," in *IET Road Transport Information and Control (RTIC) 2008 and ITS United Kingdom Members' Conference*, 2008, pp. 1–7. doi: 10.1049/ic.2008.0792.
- [13] S. Liu, Z. Wang, and H. Jiang, "Signal timing optimization with the contraflow left-turn lane design using the cell transmission model," *Transp. A Transp. Sci.*, vol. 18, no. 1, pp. 1–24, 2021, doi: 10.1080/23249935.2021.1936280.
- [14] S. Araghi, A. Khosravi, and D. Creighton, "A review on computational intelligence methods for controlling traffic signal timing," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1538–1550, 2015, doi: 10.1016/j.eswa.2014.09.003.
- [15] F. Lian, B. Chen, K. Zhang, L. Miao, J. Wu, and S. Luan, "Adaptive traffic signal control algorithms based on probe vehicle data," *J. Intell. Transp. Syst. Technol. Planning, Oper.*, vol. 25, no. 1, pp. 41–57, 2021, doi: 10.1080/15472450.2020.1750384.
- [16] W. H. Lee and C. Y. Chiu, "Design and implementation of a smart traffic signal control system for smart city applications," *Sensors*, vol. 20, no. 2, 2020, doi: 10.3390/s20020508.
- [17] M. Koziarski and B. Cyganek, "Impact of low resolution on image recognition with deep neural networks: An experimental study," *Int. J. Appl. Math. Comput. Sci.*, vol. 28, no. 4, pp. 735–744, 2018, doi: 10.2478/amcs-2018-0056.
- [18] L. Xue, X. Zhong, R. Wang, J. Yang, and M. Hu, "Low - resolution vehicle recognition based on deep feature fusion," *Multimed. Tools Appl.*, vol. 77, no. 20, pp. 27617–27639, 2018, doi: 10.1007/s11042-018-5940-6.
- [19] W. Zou and P. C. Yuen, "Very low resolution face recognition problem," in *IEEE 4th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2010*, no. May, 2010, doi: 10.1109/BTAS.2010.5634490.
- [20] W. Yang, X. Luo, B. Fang, D. Zhang, and Y. Y. Tang, "Fast and accurate vanishing point detection in complex scenes," in *2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014*, pp. 93–98, 2014, doi: 10.1109/ITSC.2014.6957672.
- [21] R. B. Wicaksono, S. Aulia, S. Hadiyoso, and B. Hidayat, "Human height and weight classification based on footprint using gabor wavelet and K-NN methods," *J. Infotel*, vol. 14, no. 2, pp. 101–107, 2022, doi: 10.20895/infotel.v14i2.728.
- [22] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *In Pract.*, vol. 7, no. 1, pp. 1–16, 2006, doi: 10.1.1.117.6808.
- [23] Matlab, "Matlab Motion-based Multiple Object Tracking," The MathWorks, Inc, 2023.
- [24] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast vanishing-point detection in unstructured environments," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 425–430, 2012, doi: 10.1109/TIP.2011.2162422.
- [25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "R-CNN: Regions with CNN features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, [Online]. Available: [http://gwylab.com/pdf/rcnn\\_chs.pdf](http://gwylab.com/pdf/rcnn_chs.pdf)
- [26] Matlab, "Matlab Train Object Detector Using R-CNN Deep Learning," The MathWorks, Inc, 2023.