International Journal of Innovative
Computing, Information and Control
Volume **16**, Number **4**, August **2020**

 ISSN 1349-4198
pp. **1353–1370**

THE RAIN NOISE REDUCTION USING GUIDED FILTER TO IMPROVE PERFORMANCE OF VEHICLE COUNTING

Budi Setiyono, Dwi Ratna Sulistyaningrum, IGN Rai Usadha and Aditya Pratama Nusantara

Department of Mathematics
Institut Teknologi Sepuluh Nopember
Kampus ITS, Sukolilo, Surabaya 60111, Indonesia
masbudisetiyono@gmail.com

Received December 2019; revised April 2020

Abstract. *The calculation of the number of vehicles on the road using digital image processing technology is expected to be faster and more accurate. Nevertheless, there are obstacles such as noise that arises due to rain because it can reduce the accuracy of the calculation of the number of vehicles. Therefore, we need a rain noise reduction process to improve the accuracy of vehicle calculations. The guided filter has the advantage of reducing noise, including snow noise and fog noise. The experiment was conducted by comparing the accuracy of the calculation of the number of vehicles in noise conditions after and before they were reduced with heavy rain, moderate rain, and drizzle data. The results show that accuracy is increasing for heavy and moderate rainfall data, while the drizzle data has decreased. Therefore, the guided filter is suitable for heavy and moderate rain conditions.*

Keywords: Vehicle counting, Rainfall, Noise reduction, Guided filter

1. **Introduction.** Smart city is an urban development and management concept by utilizing Information and Communication Technology (ICT), to connect, monitor, and control various resources within the city, to be more productive and efficient. Besides that, it is also to maximize services to its citizens and support sustainable development [1,2]. One component of a smart city is smart transportation, known as the intelligent transportation system, which includes vehicle transportation management on the highway. Traffic monitoring devices on the highway such as CCTV (Closed Circuit Television) are still not perfect without a system that can detect vehicles automatically in traffic problems, namely traffic congestion. In Indonesia, traffic congestion is one of the problems faced in big cities, for example, Jakarta, Surabaya, due to the increasing volume of vehicles.

The latest data from the *Biro PusatStatistik (BPS) Indonesia*, with the number of vehicles 121,394,185, in 2016 with the number of vehicles 129,281,079, and in 2017 reached 138,556,669 which means that the number of vehicles has increased every year [3,4]. The increase in the number of vehicles that are quite high, without being matched by infrastructure development, especially roads and traffic engineering, will have an impact on congestion. One of the essential things that can be used as a basis for traffic engineering is knowing the number of vehicles and their types passing on the highway. Vehicle calculations can be done by using CCTV cameras, which are widely installed on main roads, using video processing. However, when there is a large amount of noise, the accuracy of the vehicle calculation will decrease. Rain is one of the most common noises in tropical countries. Therefore, we need a mechanism to reduce noise due to rain, so that the vehicle

DOI: 10.24507/ijicic.16.04.1353

calculation accuracy is still good. Thus, this paper can be used as a reference for relevant agencies, for example, the transportation department, if it will automate vehicle counting through CCTV, especially in rainy conditions.

Rain is the process of falling droplets from clouds. When rain falls, more falling droplets can shorten visibility so that it can affect the accuracy of counting the moving vehicles. The vehicle observed may not be clear because it is blocked by rain. In digital video processing, rain is considered a noise that can obstruct the observation of an object. Rain causes inconvenience and problems in the process of counting moving vehicles based on video processing [5]. Therefore, we need a method in processing to reduce or even eliminate the presence of rain noise on the video to improve the accuracy of counting moving vehicles. Many researchers reduce rain noise in a single image [6-9]. Meanwhile, our study conducted rain noise reduction for video. We use guided filters because this method is good enough to reduce rain noise [6,10,11]. Previous researchers have never used guided filters to reduce rain noise, for the case of counting vehicles. While in this study, we will do it and analyze the effect of guided filters to reduce rain noise to increase accuracy in vehicle counting.

This paper is divided into several sections. The first section is an introduction. The second section is some other research related to this work. The third section is the research methodology, the steps taken by researchers to solve the problem, consisting of preprocessing, guided filters to reduce rain noise, and the process of counting vehicles. At the end of the third section, we provide a block diagram of the research, which is a summary of the stages described in the previous section. Next, the fourth section contains experiments and analysis of experimental results. Finally, the last section summarizes conclusions and points out future work.

2. **Related Works.**

Based on the object, there are studies on reducing rain noise using a single image [6-9] and video [12-14]. Liao et al. [14] put a camera inside the vehicle for video capture. Their research is to eliminate noise due to raindrops dripping on the windshield in an elliptical shape. Meanwhile, we put the camera outside the vehicle, with the camera's perspective from above, so we can detect rain based on lines. Other researchers use guided filters to reduce various noise in images and videos [9,11,12,15,16]. However, this research is only done for the object of an image, while our research is for video. There is a guided filter study for video, namely the research of Gaikwad and Scholar [12]. In their research, they compared PSNR values before and after rain noise reduction, without analyzing the effect of rain noise control on object detection. Whereas in this study, we will analyze the effect of reducing rain noise for object detection. Also, their research conducted a guided filter process as a process of eliminating rain noise with the reference image being the image itself without rain, in that study compared with the bilateral filter, which had better performance. The weight addition process is weighing in the reference image. In the study, the authors can minimize the adverse effects of guided filter, and rain removal methods are considered simple but effective. The application of guided filters for video noise removal is made by dividing the video into several frames then converting 2D input frames into 1D frames then filtering using guided filter. Researchers compared the PSNR of the guided filter and bilateral filter, resulting in a higher PSNR value for the guided filter so that, in this case, the guided filter has a better performance. However, these studies have not explained further about the influence of guided filters if applied in a rainy-noise video.

Najiya and Sreeram performed a guided filter algorithm three times in the rain removal process, including determining the low-frequency part, high-frequency part, improving the reference image by adding edge enhancement with Sobel to thicken the edge pixels reduced

by the guided filter [16]. In their research, testing is done only by visual means, without testing the quantitative accuracy of the method used. While in this work, accuracy is measured based on visual observations and quantitatively. Also, we see the success of rain noise reduction and its effect on the calculation of moving vehicles on the highway.

Moving vehicle calculation is done by separating the background and foreground. Some researchers use the Gaussian Mixture Model (GMM) because GMM's performance is quite good [17-20]. The edge detection method is used by researchers for the process of tracking, classifying, and counting moving vehicles [21]. Researchers also use pixel quantities [22] and fuzzy to detect vehicles/objects [23-25]. In their research, it was found that edge detection can detect the edges of a vehicle with extreme sensitivity. However, other detection methods are needed to recognize patterns of vehicles that have been detected by the edge detection method. We developed a system that can count the number of vehicles, and classify them in two classes, in rainy conditions. Because rain noise is quite disturbing in the process of calculating vehicles, so before counting and classification, we reduce rain noise first, using guided filter.

3. **Proposed Method.**

The main processes in this research are preprocessing rain noise reduction using guided filter, and vehicle calculation. In the following section, the authors will explain in more detail these stages.

3.1. **Preprocessing.**

Data input is a video with rain noise, and then it will be extracted into a series of frames, where each frame will be processed. Preprocessing aims to improve image quality so that it will be easier to process at a later stage. For each extracted frame, preprocessing will be done first, aiming to minimize the appearance of object shadows that can affect the shape of the object so that it interferes with the object detection process, which will also affect the counting and classification process. Preprocessing in this research is in the form of contrast and brightness enhancement. Suppose pixels $f(i, j)$ are the original pixel intensity at coordinates (i, j) and $g(i, j)$ is the resulting pixel intensities where $\alpha > 0$ is the gain parameter (contrast), and β is the bias parameter (brightness), so the contrast and brightness enhancement process is defined by Equation (1).

$$g(i, j) = \alpha f(i, j) + \beta \tag{1}$$

Increased contrast and brightness aims to clarify the difference between background and foreground, making it easier to separate the two parts.

Whereas in the background subtraction process, to separate the background with the foreground is conducted by the following process: every frame that has been extracted, looks for the background model that will be used to generate the foreground image based on the background model.

3.2. **Guided filter methods.**

Image filtering is one of the image processing operations used to suppress high frequencies in the image, such as smoothing the image or suppressing low frequencies such as clarifying or detecting edges in the image. The primary purpose of the image filtering process is to make the image appear better or appear more transparent for analysis. Figure 1 shows an illustration of a guided filter process. For image filtering to produce the expected image, a variety of image filtering studies have emerged, one of which is the guided image filtering method [11,12]. Furthermore, guided image filtering is an image filtering technique that takes two inputs, namely the image input that needs to be improved and the filter coefficient for image processing. The filter coefficient in the guided image filter method is variable and is determined using reference images. In this case, the guiding image is the input image itself.

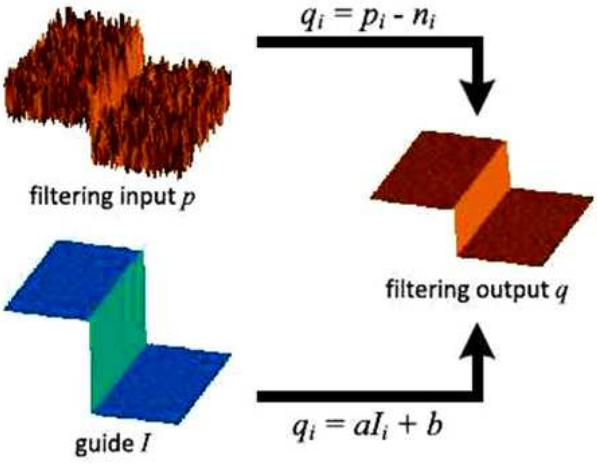


FIGURE 1. Guided filter illustration [26]

First, it is defined as the general form of the filter process where the reference image, p input image, and q output image. Furthermore, the key assumption of the guided filter is a local linear model between the guidance I and the filtering output q. We assume that q is a linear transform of I in a window ω_k centered at the pixel k:

$$q_i = a_k I_i + b_k, \quad \forall i \in \omega_k \tag{2}$$

where (a_k, b_k) are constant coefficients in ω_k. A square window with radius r is used. To determine the linear coefficients (a_k, b_k), we need constraints from the filtering input p. We model the output q as the input p subtracting some unwanted components n like noise:

$$q_i = p_i - n_i \tag{3}$$

We try to maintain the linear model (2) by minimizing the difference between q and p by minimizing the following cost function in the window ω_k. Specifically,

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2 \right) \tag{4}$$

Equation (4) is the linear ridge regression model:

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \text{ and } b_k = \bar{p}_k - a_k \mu_k \tag{5}$$

where μ_k = Mean I in ω_k, σ_k^2 = Variance I in ω_k, $|\omega|$ = Number of pixels in ω, $\bar{p}_k$ = Mean p in ω_k.

After calculating (a_k, b_k) for all ω_k in the image. Next, calculate the output filter as follows

$$q_i = \underline{a}_i I_i + \underline{b}_i$$

where:

$$\underline{a}_i = \frac{1}{|\omega|} \sum_{k \in \omega_k} a_k \tag{6}$$

$$\underline{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_k} b_k \tag{7}$$

Equations (3), (4), (5), and (6) are definitions of the guided filter so that the guided filter algorithm is obtained as follows [15].

Algorithm 1. Guided Filter

Input: input frame p, reference image I, radius r, regulation ϵ
Output: q.

1. Mean and Correlation
   $mean_I \leftarrow f_{mean}(I)$
   $mean_p \leftarrow f_{mean}(p)$
   $corr_I \leftarrow f_{mean}(I. * I)$
   $corr_{Ip} \leftarrow f_{mean}(I. * p)$
2. Variance and Covariance
   $var_I \leftarrow corr_I - mean_I. * mean_I$
   $cov_{Ip} \leftarrow corr_{Ip} - mean_I. * mean_p$
3. Coefficients a and b
   $a \leftarrow cov_{Ip}./(var_I + \epsilon)$
   $b \leftarrow mean_p - a. * mean_I$
4. Mean a and b
   $mean_a \leftarrow f_{mean}(a)$
   $mean_b \leftarrow f_{mean}(b)$
5. Output q
   $q \leftarrow mean_a. * I + mean_b$

f_{mean} is an average filter with radius r. Intuitively *corr*, *var*, *cov* in the sequence have the meaning of correlation, variance, covariance. Then ϵ is the regulatory parameter of a. Figure 2 shows an example of experimental results from the implementation of the guided filter algorithm. Figure 2(b) shows the result of the guided filter. The image looks smoother than Figure 2(a), and the sound of rain seems to decrease.

(a) (b)

FIGURE 2. Example: the results of noise reduction with guided filter: (a) input image; (b) result image

3.2.1. *Filtering.* This process applies guided filters to producing a low-frequency part of the input image to obtain a high-frequency part using Equation (8):

$$I = I_L + I_H \tag{8}$$

In the application of guided filters, the edges of the image will become very smooth. So the low-frequency part will have a smooth edge. Therefore, edge enhancement is highly recommended to increase edge sensitivity and make it almost the same as the original frame, and the result of edge enhancement is denoted by I_{LE} which becomes the reference image at the filtering stage. By using Equation (8), a high-frequency part of I_H can be obtained, which will be the input at the filtering stage. At I_H, there is the texture of the object and rain. So the filtering process removes rain and retains the texture of the object from I_H. Therefore, re-applied guided filter for I_H input with the I_{LE} reference image produces a non-rainy high-frequency part that will be used at a later stage for image.

3.2.2. *Recovering.* This step is to obtain smoother and brighter image results through the recovery mechanism. This mechanism encourages to form good results that are closer to the input image. The resulting image contains a combination of filtered I_L and I_H parts.

The resulting image is called the restored image and is denoted by I_{Rec}, obtained from (9):

$$I_{Rec} = I_L + I_H \tag{9}$$

The effect produced by the guided filter to some pixels on I_{Rec} will be blurred where the pixel in question is near the removal of rain. So the removal will be carried out again by determining the minimum pixels between I_{Rec} and the input image I notified by I_{Cr} (10).

$$I_{Cr} = Min(I_{Rec}, I) \tag{10}$$

After using the guided filter, the value of the deleted rain part becomes a little higher than the value of the adjacent pixels. That makes the display less optimal. Therefore, the sum is done as in Equation (11). The results obtained are called refinement of the reference image and denoted by I_{Ref}. The next guided filter will use it as a reference image. The sum in question is:

$$I_{Ref} = \alpha I_{Cr} + (1 - \alpha) I_{Rec} \tag{11}$$

with $\alpha = 0.8$. In Figure 3, we show the steps of the guided filter described above and the results, if applied as an image.

3.3. **Vehicle counting.** Separation of background and foreground in the image is using the Gaussian Mixture Model (GMM) method.

3.3.1. *Background subtraction using GMM method.* GMM is used in this process because this method is resistant to changes in the characteristics of the background model, which may change at any time. GMM models will form pixel color data based on time. The model produces two parts, namely the background model and the non-background model. The background model is a model that reflects the background. Each pixel will be grouped by distribution, which is considered the most effective as a background model. The higher the standard deviation value, the stronger the smoothing that occurs in the image. Each pixel has its model. The processed data is the intensity of the pixels obtained from the input frame. Every frame is extracted; the model for each pixel will be updated. In the system to be built, several parameters have been defined previously. This parameter is used for background subtraction processes with GMM [5]. These parameters include α (learning rate) with a value of 0.01, the number of Gaussian components, namely 3, T (threshold) with a value of 0.4. There is also an initial initialization of some GMM parameters, among others: ω_k which is the weight of each pixel in the k-Gaussian with a value of 1/3 where 3 is the sum of the Gaussian distribution; μ_k is the mean of each pixel in the k-Gaussian where each pixel of each Gaussian has a random value between 0 and 255; σ_k is the standard deviation for each pixel in k-Gaussian.

3.3.2. *Object detection and labelling.* To obtain good results in object detection, in this study, we used three processes, namely smoothing, shadow removal, and morphology, namely erosion and dilation. Erosion is used to thin objects in a binary image, as shown in Figure 5. Dilation is useful for expanding or thickening objects in binary images. Figure 6 shows an example of widening. Smoothing is a process to improve the image so that the noise in the foreground image results in a background subtraction process.

The background subtraction process used in this study can simultaneously be used to detect shadows. However, to maximize the detection process, the shadows that are identified are not immediately removed when they are detected but require an alignment process so that the suspected shadow portion becomes accurate. Therefore, the shadow removal process in this system is carried out after the smoothing process. The results of the image from the smoothing process have three levels of gray, namely 0, 127, and 255.

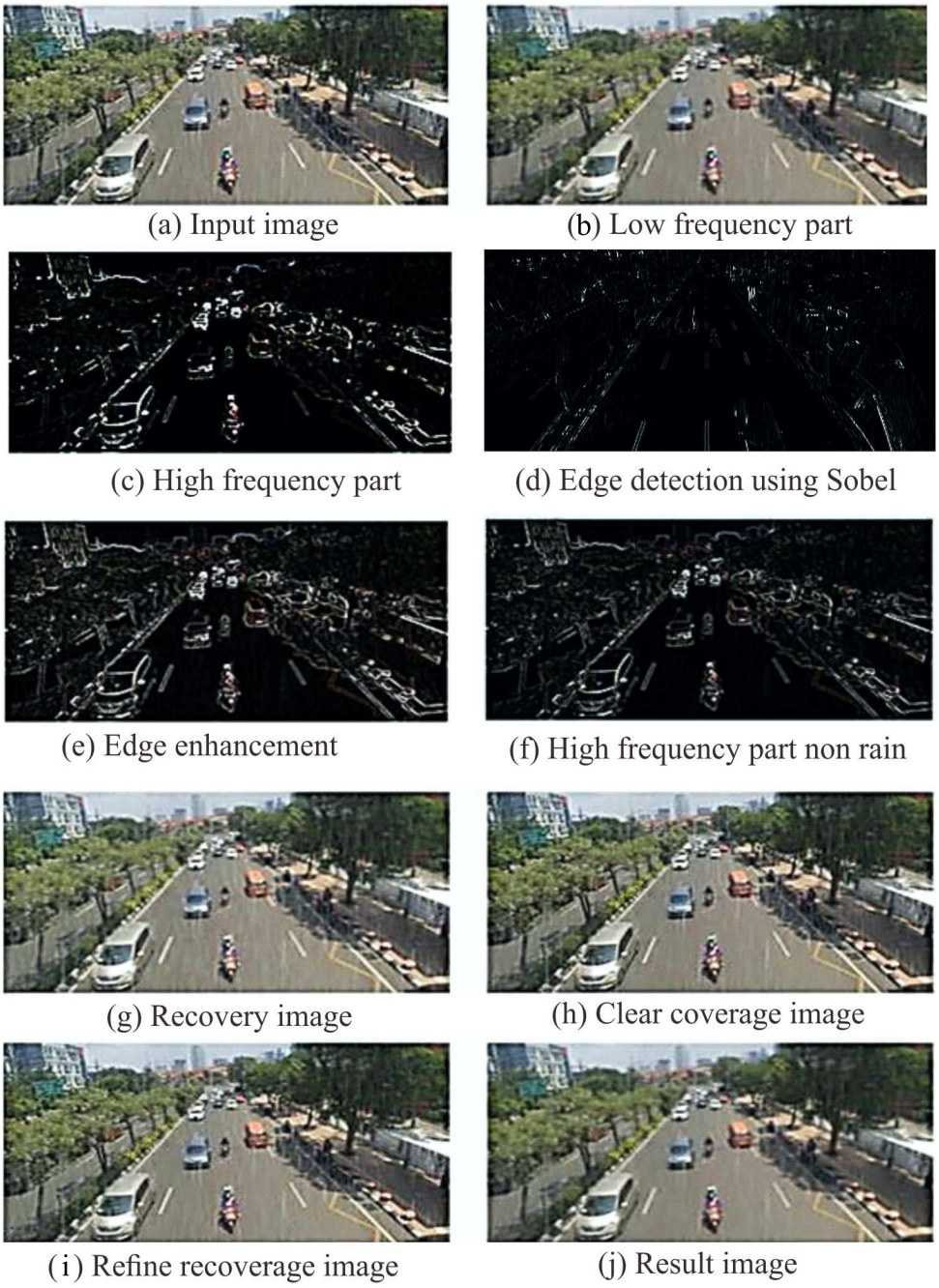


FIGURE 3. The results of the procedure implementation

Shadow removal is a process where the value is mapped to binary numbers, 0 and 1, as in Equation (12).

$$I_{bin}(x,y) = f(x) = \begin{cases} 1, & I_{gray}(x,y) = 255 \\ 0, & I_{gray}(x,y) = 0 \text{ or } I_{gray}(x,y) = 127 \end{cases} \tag{12}$$

Figure 4 shows a shadow removal process, the pixel which was initially gray, with the threshold process will be removed to black, so it does not obscure the object to be detected. Object detection is done after the image undergoes a morphological process.

In Figures 5 and 6, if they are combined sequentially, they become closed morphological processes.

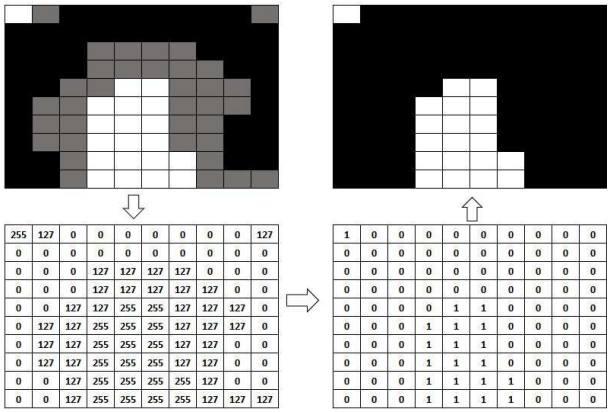


FIGURE 4. Shadow removal

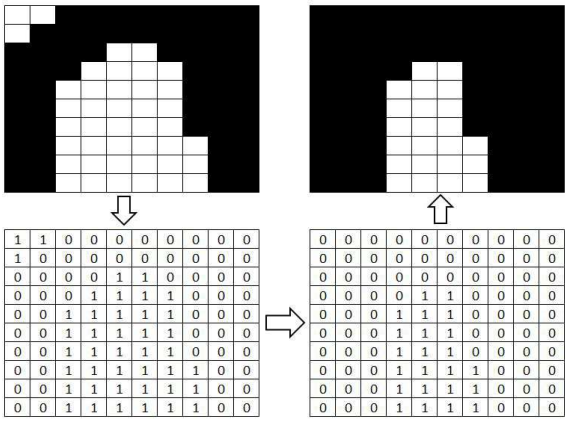


FIGURE 5. Erosion

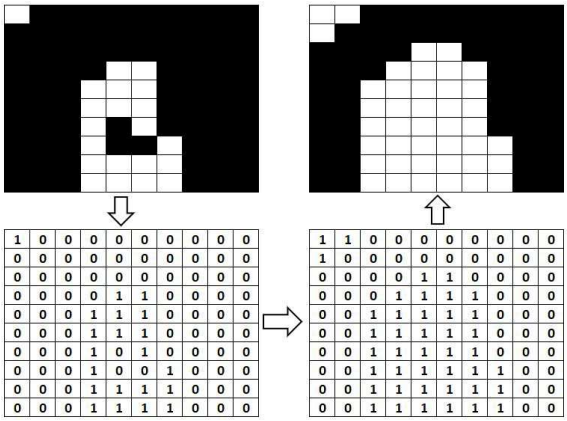


FIGURE 6. Dilation

The object detection process in this study uses the concept of connected components to detect contours in binary images. Contours that are detected on an object are based on the outermost border with the following method. The border following algorithm 2 is as follows [28].

Algorithm 2. Border Following Algorithm

1. The scanning process runs with a raster scan, which is from top to bottom (row), left to right (column).
2. Look for pixels $f(i,j)$ as candidates for outer border pixels that meet the conditions $f(i,j-1) \leftarrow 0$, $f(i,j) \leftarrow 1$ and if the pixel is the outermost border, which meets the following conditions:
   i. The pixels $f(i,1), f(i,2), \ldots, f(i,j-1)$ are pixels 0, or
   ii. Pixels $f(i,h)$ is parallel to the border point, and pixels $f(i,h+1)$ are the background
   iii. Label as a border.
3. If the border boundary has been found, namely $f(i,j-1) \leftarrow 1$ and $f(i,j) \leftarrow 0$, then label it as a border mark.
4. Run raster scanning until all pixels are passed.

An example of a border following in a binary image can be seen in Figure 7 with the border given Label 2.

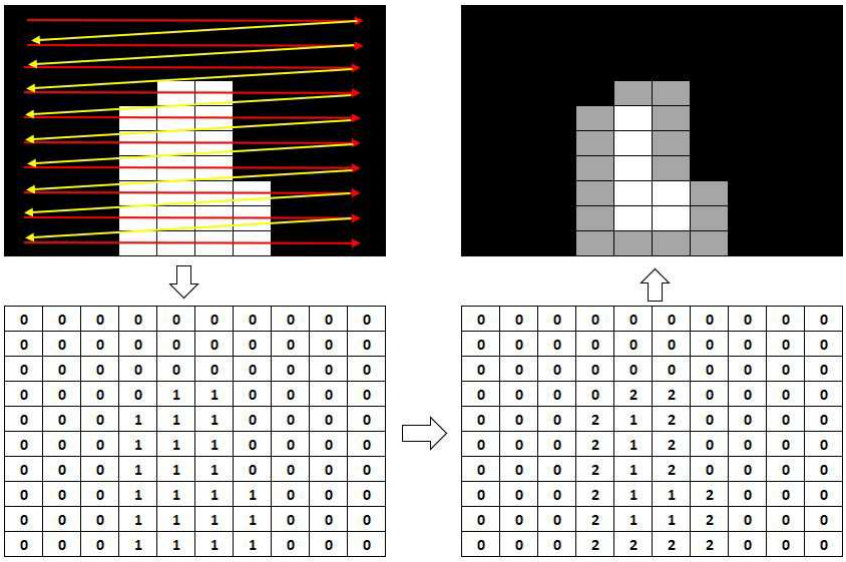


FIGURE 7. Border following

ROI is an area where vehicles will be labeled and classified. The labeling process starts when the detected object enters ROI. Labels will always be updated and matched in the following frames until the object exits ROI. When the object is in ROI, the position of the vehicle between frames will always be recorded, and when the object leaves the ROI will be calculated. The stored data is displayed as a calculation of the number of vehicles. To ensure that objects are not counted and are stored more than once, the process of tracking objects between frames needs to be done. Tracking is a process that aims to find the same object in the previous frame. Two objects are said to be the same if they satisfy Inequalities (13) and (14).

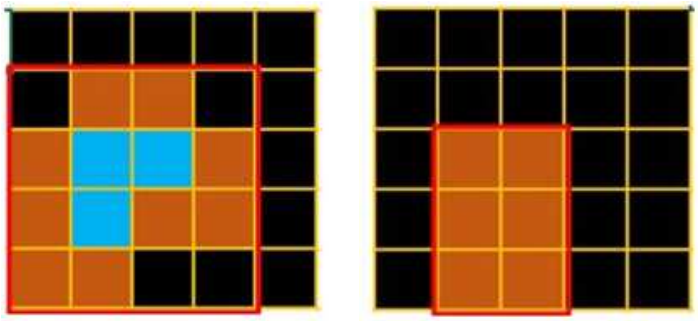
$$L_{object1} \leq \frac{L_{object2} + R_{object2}}{2} \leq R_{object1} \tag{13}$$

$$T_{object2} \leq T_{object1} \leq B_{object2} \tag{14}$$

where $object1$ is the object being processed, $object2$ is an object in the previous frame, L is the left border of the object, R is the right border of the object, T is the upper limit of the object, and B is the lower limit of the object. If the two objects are identified the same, then the ID on the previous object will move to the object being processed and do the calculation and classification of the vehicle. The process will continue to apply until the object leaves ROI.

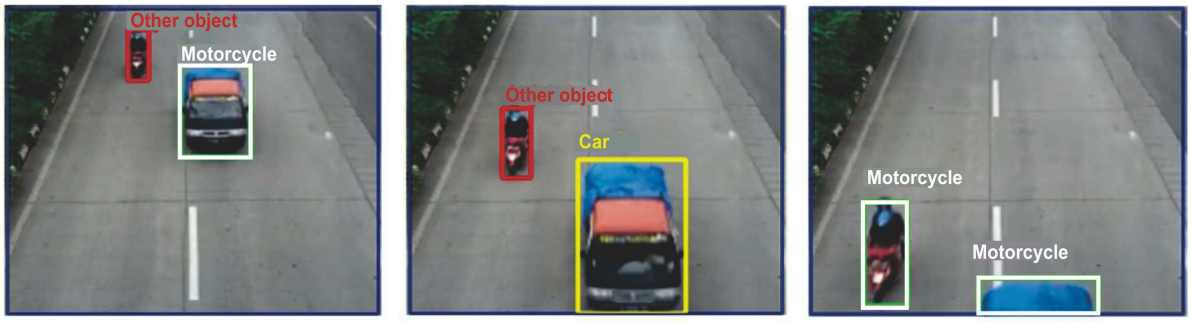
3.3.3. *Tracking, classification, and counting.* Furthermore, vehicles that have an ID will be classified and calculated according to the type of vehicle. The vehicles are classified into two types, namely motorcycles and cars. The process of classifying vehicle types is based on the result of the BLOB analysis. Motorcycle and car sampling need to be done to calculate the area of the bounding box so that a common area for motorcycles and cars is obtained. In order to distinguish motorbike and car, we use the number of pixels in the object. Figure 8 shows an illustration of a bounding box, which in picture (a) has area 16, and picture (b) has area 6. For example, the selected threshold is 11; the left image is classified as a car because it has an area larger than the threshold, while the picture (b) which has an area less than the threshold is classified as motorcycles.

Figure 9 shows the result of vehicle classification and the provision of bounding boxes in multi vehicles. Because the system that is built must be able to count multi-object



(a) (b)

FIGURE 8. The threshold for the number of pixels as a basis for vehicle classification: (a) car; (b) motorcycle



(a) (b) (c)

FIGURE 9. Bounding box and ID on multi vehicle detection

vehicles and always record the vehicle when in ROI, it takes a list to store vehicle objects that are detected when the vehicle is on ROI, namely *listObj*, as seen in Algorithm 3.

Algorithm 3. Tracking and Counting

1. *listObj* ←{}
2. ***if*** object in ROI area
   repeat (read frame)
   if (detObj and ID == ID in prev Frame)
   listObj ← ID; update location of the bounding box
   else
   listObj ← new ID
   until (last frame or object out of ROI)
3. Classify according to the threshold of cars and motorcycles
4. Calculate the amount respectively

Initially, *listObj* is empty; the vehicle detected is a new object which is immediately given an ID label, entered into *listObj*. In the next frame, if the detected vehicle is the same as the vehicle in the previous frame in *listObj*, the object updating process will be carried out, i.e., an updating ID on the new vehicle object, updating the bounding box location.

If the vehicle is not the same or does not match the vehicle objects that are in the *listObj*, it can be said that the detected vehicle is new, given an ID, and entered in the

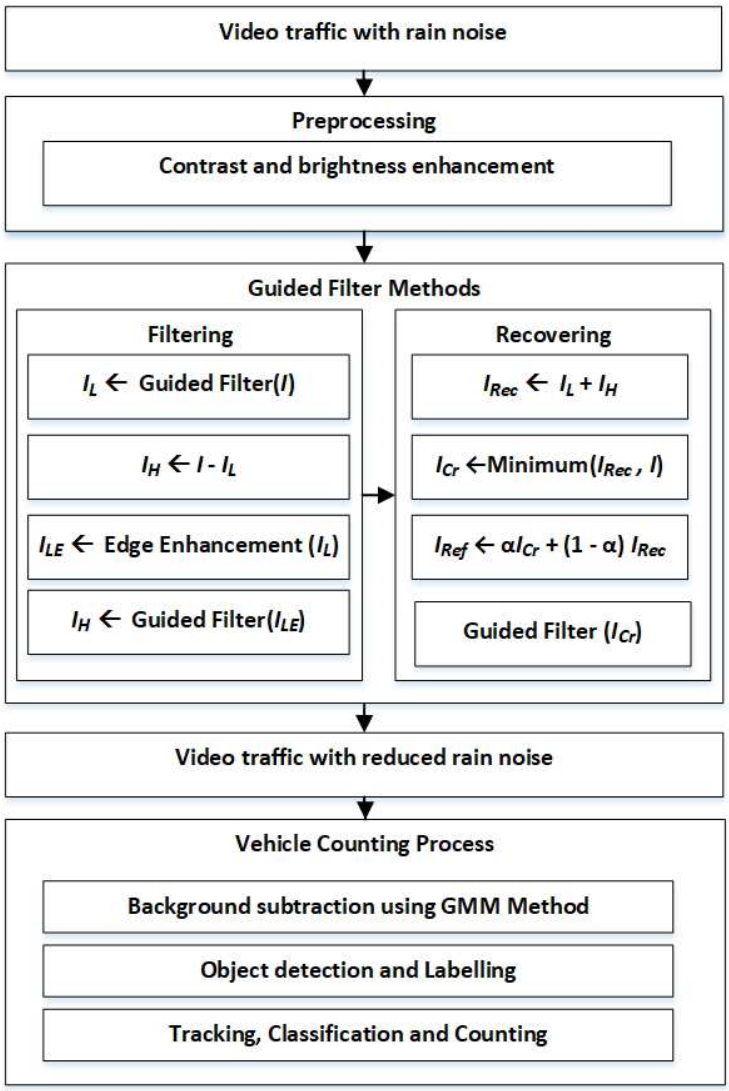


FIGURE 10. The stages of the proposed method

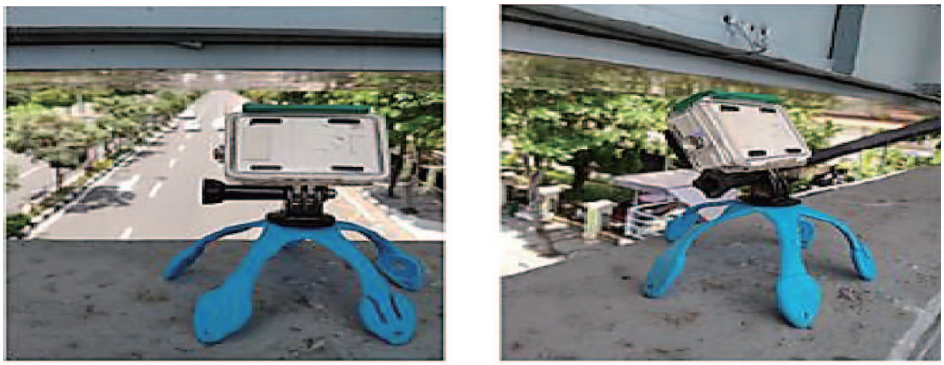
listObj. This process is carried out as many objects as are detected in ROI. When a vehicle exits ROI, what is done is to calculate the area of the bounding box, then classify it between the motorbike or the car based on the area and then calculate every object that comes out. Overall, the process steps outlined above can be seen in the block diagram shown in Figure 10.

4. **Experiments and Results.** Video acquisition is carried out on a pedestrian overpass, with a motionless camera. Figure 11 explains how the camera is positioned during video acquisition.

The trial run of the program was carried out on videos that had been stored in computer storage obtained from taking over the Surabaya pedestrian overpass. Table 1 shows the video file that will be used with the actual number of vehicles.

4.1. **Testing the effect of using guided filter.** Existing data is video data without rain, which then generates rain noise. The data are used in the form of heavy rain, moderate, and low static background moving objects.

The parameters of heavy rain, moderate, and low are based on the parameters of the tool after effect. Heavy rain: Raindrops 20,000/second, the size of the rain is 6 mm,



(a) (b)

FIGURE 11. Camera position during video acquisition: (a) rear view; (b) side view

TABLE 1. Data experiment

No	File	Origin	
		Motor	Mobile
1	Diponegoro31d.mp4	84	14
2	Pemuda27d.mp4	51	10
3	Suramadu39d.mp4	13	5
4	SuramaduB39d.mp4	15	6
5	Wonokromo.mp4	51	8

FIGURE 12. Heavy rain, moderate rain, drizzle

speed 4000. Moderate Rain: 10,000 raindrops/second, the size of the rain is 4 mm, speed 3000. Dizzle: Raindrops 5,000/second, the size of the rain is 3 mm, speed 2000. The rain generated results are then added to the vehicle traffic video, shown in Figure 12. Figure 12 shows the visual appearance of various characteristics of rain generated by **specific** parameters.

The more rain that is raised, the higher the noise will appear. Furthermore, the video that has been added with various characteristics of rain is stored in a file. The first test was conducted to determine the effect of guided filters on the background subtraction, as seen in Figure 13. Figure 13 shows an example of the results of background subtraction using GMM. In Figure 13(b), it can be seen that with the application of guided filters, the effects of rain noise can be reduced well. The next experimental process is to test the performance of rain noise reduction using a guided filter.

The test is done by calculating the PSNR, by comparing the frames that are given rain noise with frames without rain noise: (i) without reducing rain noise; (ii) by reducing rain noise. Experiments are carried out for drizzle, moderate rain, and heavy rain. The test results can be seen in Figure 14. During moderate and heavy rain, Figures 14(b) and 14(c), all frames experience an increase in PSNR, but for drizzle, in Figure 14(a), some frames actually decline in PSNR. This is caused by noise that is not too large; reduction

(a) (b)

FIGURE 13. Background subtraction (a) without rain reduction and (b) with rain reduction

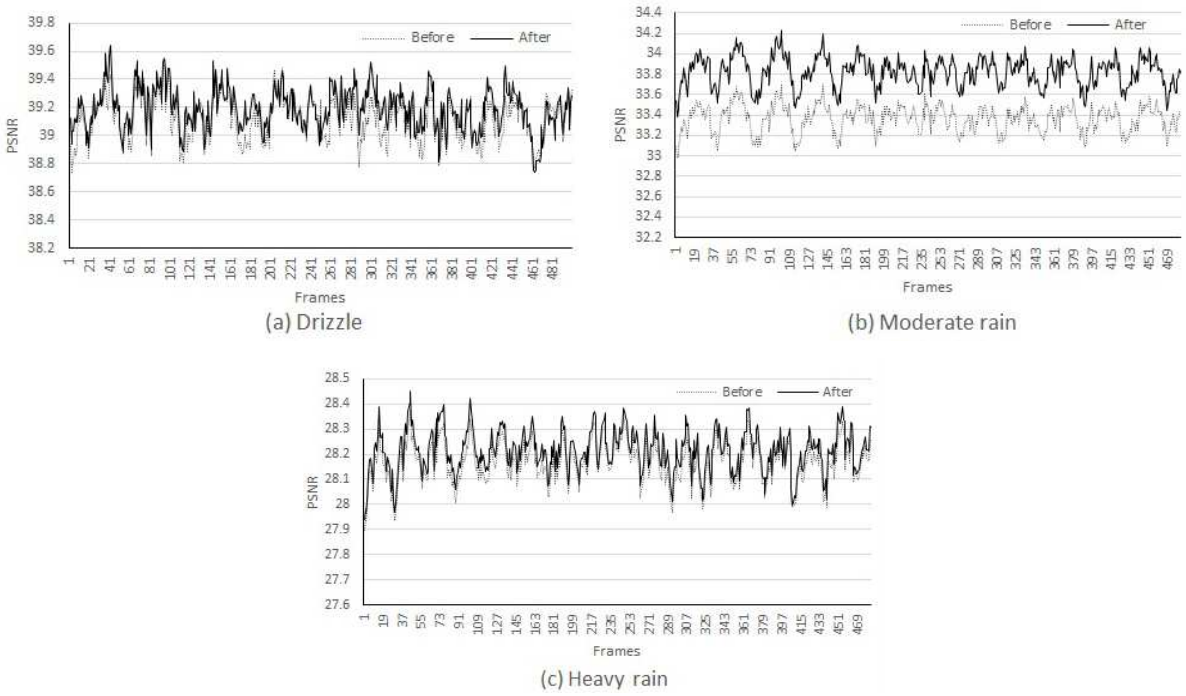


FIGURE 14. Effect of rain noise reduction on the increase in PSNR

using a guided filter will result in a blur in the image, so the separation of background and foreground becomes more difficult than a contrast image. Thus, instead, there was a decline in the value of PSNR.

4.2. **Performance classification and vehicle counting testing.** The results of vehicle classification and calculation are compiled into the results table concerning the paper. The data in the results column is obtained from the calculation results by the program that has been made. Classification performance is shown in the recall column and the precision column. The recall column shows the percentage of positive category data that were classified correctly by the system [29]. The precision column shows the percentage of the total correctly classified positive category data divided by the total positive classified data.

$$Recall = \frac{TP}{GT} \times 100\% \tag{15}$$

$$Precision = \frac{TP}{DV - MC} \times 100\% \tag{16}$$

$$Error\ rate = \frac{FP}{GT} \times 100\% \tag{17}$$

$$Detection\ rate = \left(1 - \frac{FN}{GT}\right) \times 100\% \tag{18}$$

where TP (True Positive) is the object of the vehicle that was detected correctly, DV (Detected Vehicles) is the number of vehicles detected by the system, MC (misclassified) is the number of errors in the classification, FP (False Positive) indicates the error calculation of the vehicle, such as not an object which is detected as an object and the object is counted twice. FN (False Negative) is an undetectable vehicle. GT (Ground Truth) is the actual number of vehicles. Table 2 is intended to show the values of GT (Ground Truth), DV (Detected Vehicles), TP (True Positive), MC (misclassified), FP (False Positive), and FN (False Negative).

Table 2. The value of GT, DV, TP, MC, FP, and FN

File	GT		DV		TP		MC		FP		FN	
	Mtr	Car	Mtr	Car	Mtr	Car	Mtr	Car	Mtr	Car	Mtr	Car
Diponegoro21d-deras.mp4	84	14	287	78	77	11	7	3	203	64	0	0
Diponegoro21d-sedang.mp4	84	14	106	17	80	12	4	2	22	3	0	0
Diponegoro21d-rendah.mp4	84	14	87	16	81	12	3	2	3	2	0	0
Pemuda27d-deras.avi	51	10	300	120	42	8	9	2	249	110	0	0
Pemuda27d-sedang.avi	51	10	87	30	43	8	8	2	36	20	0	0
Pemuda27d-rendah.avi	51	10	51	14	51	10	0	0	0	4	0	0
SuramaduB34d-deras.avi	13	5	383	55	12	4	1	1	370	50	0	0
SuramaduB34d-sedang.avi	13	5	64	15	12	4	1	1	51	10	0	0
SuramaduB34d-rendah.avi	13	5	39	17	12	4	1	1	26	12	0	0
Suramadu-deras.mp4	15	6	211	56	14	4	1	2	196	50	0	0
Suramadu-sedang.mp4	15	6	117	36	14	5	1	1	102	30	0	0
Suramadu-rendah.mp4	15	6	51	10	15	5	0	1	36	4	0	0
Wonokromo30d-deras.avi	51	8	125	15	48	5	3	3	74	7	0	0
Wonokromo30d-sedang.avi	51	8	59	12	51	5	0	3	8	4	0	0
Wonokromo30d-rendah.avi	51	8	54	8	50	8	0	0	3	0	1	0

Furthermore, it will be shown the comparison of the recall value, precision, and accuracy of detection before and after the rain deduction using guided filters, and the results are shown in Table 3.

TABLE 3. The value of recall and precision before and after rain reduction

No	Rain type	Recall (%)				Precision (%)				Acuracy of detection (%)			
		Before		After		Before		After		Before		After	
		Mtr	Car	Mtr	Car	Mtr	Car	Mtr	Car	Mtr	Car	Mtr	Car
1	Heavy	90.76	73.55	96.47	85.74	18.22	15.59	37.98	53.51	100.00	100.00	98.98	92.50
2	Moderate	93.04	78.31	99.37	86.67	50.08	41.40	59.65	80.49	100.00	100.00	100.00	90.00
3	Drizzle	98.89	89.81	96.93	62.33	70.00	67.54	69.68	90.59	99.61	100.00	95.48	72.67

Figure 15 shows the graphical representation of the data in Table 3. For a recall in Figure 15(a), it appears that the recall process after rain reduction has increased. This means the number of correct classifications carried out by the system compared to the actual classifications, both for motorbikes and cars, has increased. For heavy rainfall, the increase is smaller than moderate rainfall. Figure 15(b) shows an increase in the value of precision. Based on Figure 16(b), the accuracy of the system in classification increases in the type of heavy and moderate rain. Performance precision in heavy rain increased by 19.76% for motorcycles and 37.93% for cars. Performance precision at moderate rain type increased by 9.57% for motorcycles and 39.10% for cars. While there is a decrease in precision performance on drizzle type by 0.32% for motorcycles and an increase in performance by 23.05% for cars.

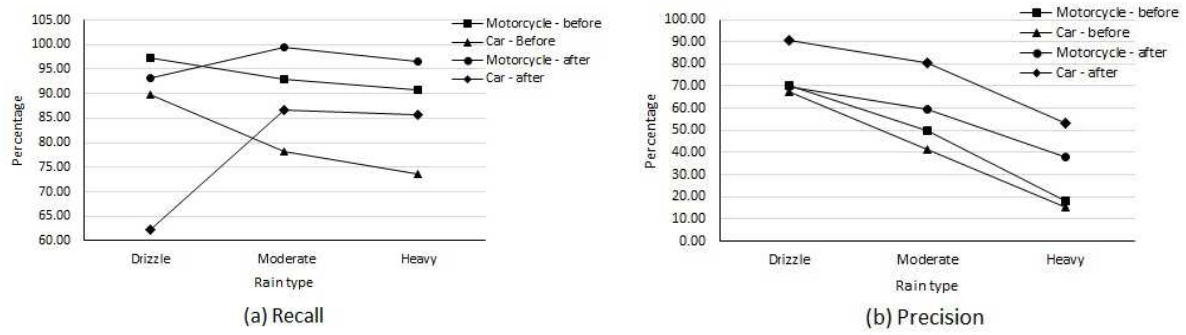


FIGURE 15. Graph of recall and precision values

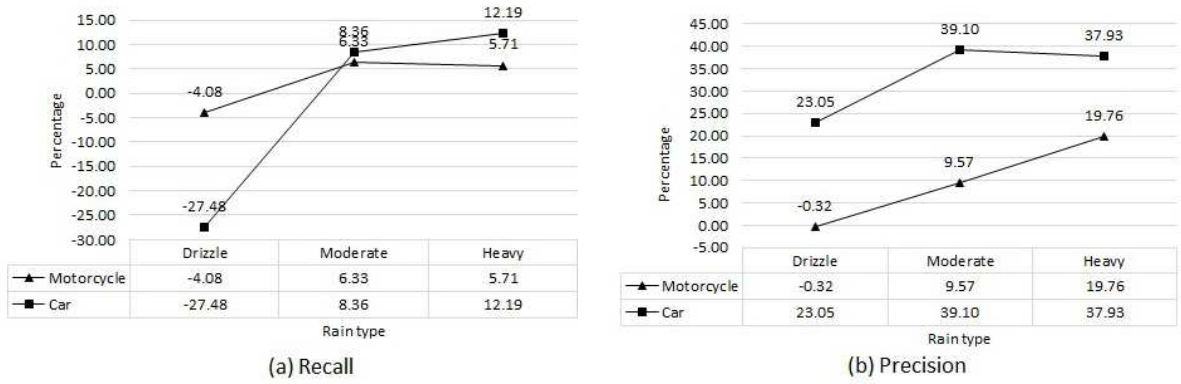


FIGURE 16. Improved recall and precision performance

Based on Figure 16(a), object detection accuracy is correctly increased for heavy and moderate rainfall types. Recall performance on heavy rain increased by 5.71% for motorcycles and 12.19% for cars. Recall performance on moderate rain increased by 6.33% for motorcycles and 8.36% for cars. While there was a decrease in recall in the drizzle type by 4.08% for motorcycles and 27.48% for cars. Low-noise rain videos have excellent video quality.

Therefore, recall and precision performance decrease in low-type rain because guided filters, which are a method of refinement, can cause objects other than rain to be affected by blurring or smooth effects, which make it difficult for the system to detect well. While the increase in vehicle counting performance after a guided filter can be seen in Figure 17.

In Figure 17(a) object detection and classification uses a wide area, while Figure 17(b) uses the number of pixels. In both object detection methods, there is a significant increase in performance related to counting accuracy, compared to before the reduction and after the reduction of rain noise.

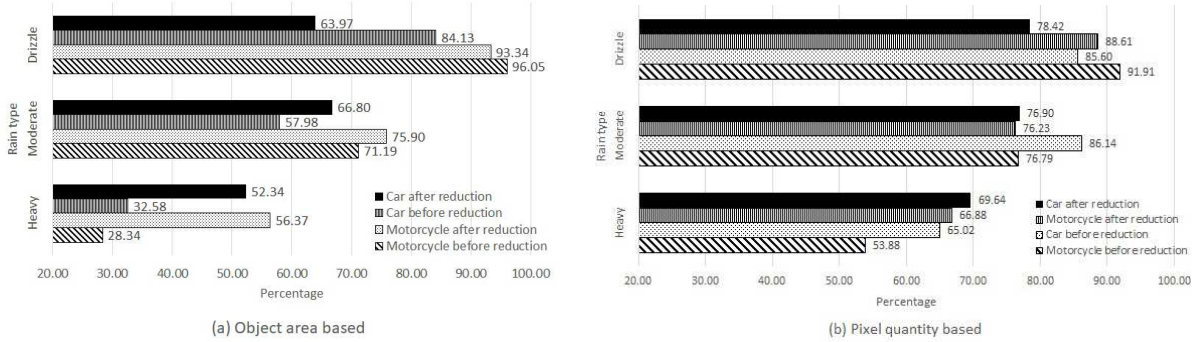


FIGURE 17. Improved counting performance

5. **Conclusion and Future Work.** This research succeeded in increasing the accuracy of counting on the type of heavy and moderate rain. However, it has not been successful in increasing accuracy for the light rain type (drizzle), because the video has a drizzle, and there is quite a bit of rain noise, so the guided filter causes the image to blur. This will complicate the processing of object detection so that the recall performance and precision decreases. This causes the vehicle object to be challenging to detect by the system accurately. From the description above, it can be concluded, guided filters can be used to improve the accuracy of detection and classification, especially in dense and moderate rain types, but not suitable for drizzle.

In subsequent studies, we will improve the accuracy of counting and design algorithms that are adaptive to changes in rainfall and changes in lighting intensity. Besides that, we also seek for a method so that for light rain, the performance can also be improved.

Acknowledgment. This work is partially supported by PDUPT with the title of Robust Tracking and Counting Systems Against Noise of Moving Vehicle based on Digital Image Processing to Support Smart Transportation Management, contract number 879/PKS/ITS/2019, 29 March 2019 (Sistem Robus terhadap Noise pada Tracking dan Counting Kendaraan Bergerak Berbasis Pengolahan Citra Digital sebagai Pendukung Manajemen Transportasi Cerdas, contract number 879/PKS/ITS/2019 Tanggal 29 Maret 2019), from the Ministry of Research, Technology and Higher Education of the Republic of Indonesia. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] V. Albino, U. Berardi and R. M. Dangelico, Smart cities: Definition, deminsion, and performance, *J. Urban Technol.*, vol.22, pp.3-21, 2015.

[2] K. Iqbal, M. A. Khan, S. Abbas, Z. Hasan and A. Fatima, Intelligent transportation system (ITS) for smart-cities using Mamdani fuzzy inference system, *International Journal of Advanced Computer Science and Applications*, vol.9, no.2, 2018.

[3] B. P. Statistik, *Statistics of Land Transportation (Statistik Transportasi Darat) 2016*, p.84, 2017.

[4] Traffic Corps Indonesian National Police (T. C. I. N. Police), The development of the number of vehicles based on the type (Perkembangan jumlah kendaraan bermotor menurut jenis), 1949-2018, *BPS*, https://www.bps.go.id/linkTableDinamis/view/id/1133, 2018.

[5] I. Gulati and R. Srinivasan, Image processing in intelligent traffic management, *Int. J. Recent Technol. Eng.*, vol.8, no.2, pp.213-218, 2019.

[6] Z. Shi, Y. Li, C. Zhang, M. Zhao, Y. Feng and B. Jiang, Weighted median guided filtering method for single image rain removal, *EURASIP J. Image Video Process.*, vol.35, 2018.

[7] Y. Luo, Y. Xu and H. Ji, Removing rain from a single image via discriminative sparse coding, *IEEE Trans. Pattern Anal. Mach. Intell.*, pp.3397-3405, 2013.

[8] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding and J. Paisley, Removing rain from single images via a deep detail network, *Proc. of the 30th IEEE Conf. Comput. Vis. Pattern Recognition (CVPR 2017)*, pp.1715-1723, 2017.

[9] J. Xu, W. Zhao, P. Liu and X. Tang, Removing rain and snow in a single image using guided filter, *IEEE Int. Conf. Comput. Sci. Autom. Eng.*, vol.2, no.2, pp.304-307, 2002.

[10] C. H. Bahnsen and T. B. Moeslund, Rain removal in traffic surveillance: Does it matter?, *IEEE Trans. Intell. Transp. Syst.*, vol.20, no.8, pp.2802-2819, 2019.

[11] X. Zheng, Y. Liao, W. Guo, X. Fu and X. Ding, Single-image-based rain and snow removal using multi-guided filter, in *Neural Information Processing. ICONIP 2013. Lecture Notes in Computer Science*, M. Lee, A. Hirose, Z. G. Hou and R. M. Kil (eds.), Berlin, Heidelberg, Springer, 2013.

[12] A. S. Gaikwad and P. G. Scholar, Filtering of video using guided filter, *Int. J. Sci. Technol. Eng.*, vol.2, no.11, pp.848-851, 2016.

[13] J. Bossu, N. Hautière and J. P. Tarel, Rain or snow detection in image sequences through use of a histogram of orientation of streaks, *Int. J. Comput. Vis.*, vol.93, no.3, pp.348-367, 2011.

[14] H. C. Liao, D. Y. Wang, C. L. Yang and J. Shin, Video-based water drop detection and removal method for a moving vehicle, *Inf. Technol. J.*, vol.12, no.4, pp.569-583, 2013.

[15] K. He, J. Sun and X. Tang, Guided image filtering, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.35, no.6, pp.1397-1409, 2013.

[16] C. Najiya and S. Sreeram, Single image rain removal using guided filter, *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol.7782, no.5, pp.135-142, 2015.

[17] B. Setiyono, D. R. Sulistyaningrum, Soetrisno and D. W. Wicaksono, Multi-vehicle speed detection using Euclidean distance based on video processing, *Int. J. Comput.*, vol.18, no.4, pp.431-442, 2019.

[18] D. Reynolds, Gaussian mixture models, *Encycl. Biometric Recognit.*, vol.31, no.2, pp.1047-1064, 2008.

[19] B. Setiyono, D. R. Sulistyaningrum, Soetrisno and H. Al-Habib, Improvement of sub-region matching illumination transfer in hybrid shadow removal method for moving vehicle video, *Int. J. Eng. Technol.*, vol.7, no.4, 2018.

[20] T. Huang, H. Peng and K. Zhang, Model selection for Gaussian mixture models, *arXiv:1301.3558v1*, 2013.

[21] Y. Li, Z. Li, H. Tian and Y. Wang, Vehicle detecting and shadow removing based on edged mixture Gaussian model, *IFAC Proc. Vol.*, vol.44, no.1, pp.800-805, 2011.

[22] B. C. Putra, B. Sctiyono, D. R. Sulistyaningrum, Soetrisno and I. Mukhlash, Moving vehicle classification using pixel quantity based on Gaussian mixture models, *The 3rd International Conference on Computer and Communication Systems (ICCCS 2018)*, 2018.

[23] K. Kobayashi, K. C. Cheok and K. Watanabe, Estimation of absolute vehicle speed using fuzzy logic rule-based Kalman filter, *Proc. of 1995 American Control Conference (ACC'95)*, pp.3086-3090, 2005.

[24] H. Pazhoumand-Dar, Object speed estimation by using fuzzy set, *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol.4, no.4, pp.241-244, 2010.

[25] J. Pouramini and A. Saeedi, Fuzzy model identification for intelligent control of a vehicle speed limit, *J. Math. Comput. Sci.*, vol.2, no.2, pp.337-347, 2011.

[26] V. Patil, P. Malathi and M. Sharma, Edge preservation using guided image filter technique, *Int. J. Eng. Res. Gen. Sci.*, vol.3, no.4, pp.498-502, 2015.
[27] N. R. Draper and H. Smith, *Applied Regression Analysis*, 3rd Edition, John Wiley & Sons, Inc., 1998.
[28] S. Suzuki and K. Be, Topological structural analysis of digitized binary images by border following, *Comput. Vision, Graph. Image Process.*, vol.30, no.1, pp.32-46, 1985.
[29] A. Crouzil, L. Khoudour, P. Valiere and D. N. T. Cong, Automatic vehicle counting system for traffic monitoring, *J. Electron. Imaging*, vol.25, no.5, p.051207, 2016.