Available online at :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Accreditated SINTA "2" Kemenristek/BRIN, No. 85/M/KPT/2020

UNIVERSITAS AMIKOM PURWOKERTO

Detection and Classification of Vehicles on the Bekasi Toll Road Using the Gaussian Mixture Models Method and Morphological Operations

Rifki Kosasih[1], Hidayat Taufik Akbar[2]

[1]Computational Mathematics Study Center, Gunadarma University
[2]Informatics Engineering, Gunadarma University
Depok, Indonesia
E-mail: rifki_kosasih@staff.gunadarma.ac.id[1], hid.taufik99@gmail.com[2]

ARTICLE INFO

History of the article:
Receive July 02, 2021
Revised December 10, 2021
Received February 22, 2022

Keywords:
CCTV
Background Substraction
Gaussian Mixture Models
morphological operations

Correspondece:
E-mail:
rifki_kosasih@staff.gunadarma.ac.id

ABSTRACT

Traffic surveillance was initially carried out directly using CCTV, but this kind of surveillance was not possible for a full day by the security forces. In addition, with the increasing growth of vehicles in Indonesia, a method is needed that can be used to assist security forces in monitoring traffic such as detecting and automatically counting the number of vehicles. Therefore, in our research, we propose a method that can detect vehicles, and count the number of vehicles from video recordings on the Bintara Bekasi toll road using background substraction methods such as gaussian mixture models and morphological operations. The results showed that the vehicle detection accuracy rate was 86.3636%, the precision was 89.0625%, and the recall was 96.6101%. In this study, vehicle classification was also carried out based on the detection results into two types of vehicles, namely cars and trucks. From the results of the research, the classification accuracy rate was obtained at 85.9649%.

INTRODUCTION

Traffic conditions in Indonesia, especially in big cities, with the example of the case in Bekasi city have begun to be monitored by CCTV which is used to monitor traffic, and in several places it is used to monitor the number and types of vehicles passing through a road. Traffic surveillance using CCTV is usually carried out directly by security forces, however, it is impossible to supervise the surveillance for a full day by the security forces. Therefore we need a method that can be used to assist security in monitoring traffic, such as detecting and counting the number of vehicles automatically.

Several studies have been conducted in the field of vehicle detection such as Rad using mean filters to get the background and foreground from the video. In his research using the mean filter, it was found that the vehicle detection accuracy rate was 73.1333% (Rad et al., 2010). Next, Qing Tian used the Support Vector Machine (SVM) and used Histograms of Oriented Gradiens (HOG) to extract the background. In his research, Qing Tian detected a vehicle in a traffic condition that was not heavy so that it obtained an accuracy rate of 97.07% (Tian et al., 2013). Kosasih used the Eigen Background generated from the Principle Component Analysis (PCA) to get the background. In his research, the detection was carried out when the traffic was not heavy so that an accuracy rate was obtained of 95% (Kosasih et al., 2020). Next, Kosasih uses running average background substraction to get the background and foreground of a vehicle video. In his research, the detection was performed when traffict was heavy, so that an accuracy rate was 80.43% (Kosasih & Arfiansyah, 2020).

In previous studies, several researchers conducted vehicle detection in conditions that were not crowded and had not classified the type of vehicle. In this research, we propose a method that can detect, classify vehicles, and count the number of vehicles from the video recordings, namely the background substraction method using Gaussian Mixture Models (GMM) and morphological operations.

RESEARCH METHODS

In this chapter we discuss the research stages of vehicle detection using Gaussian Mixture Models background substraction which can be seen in Figure 1.

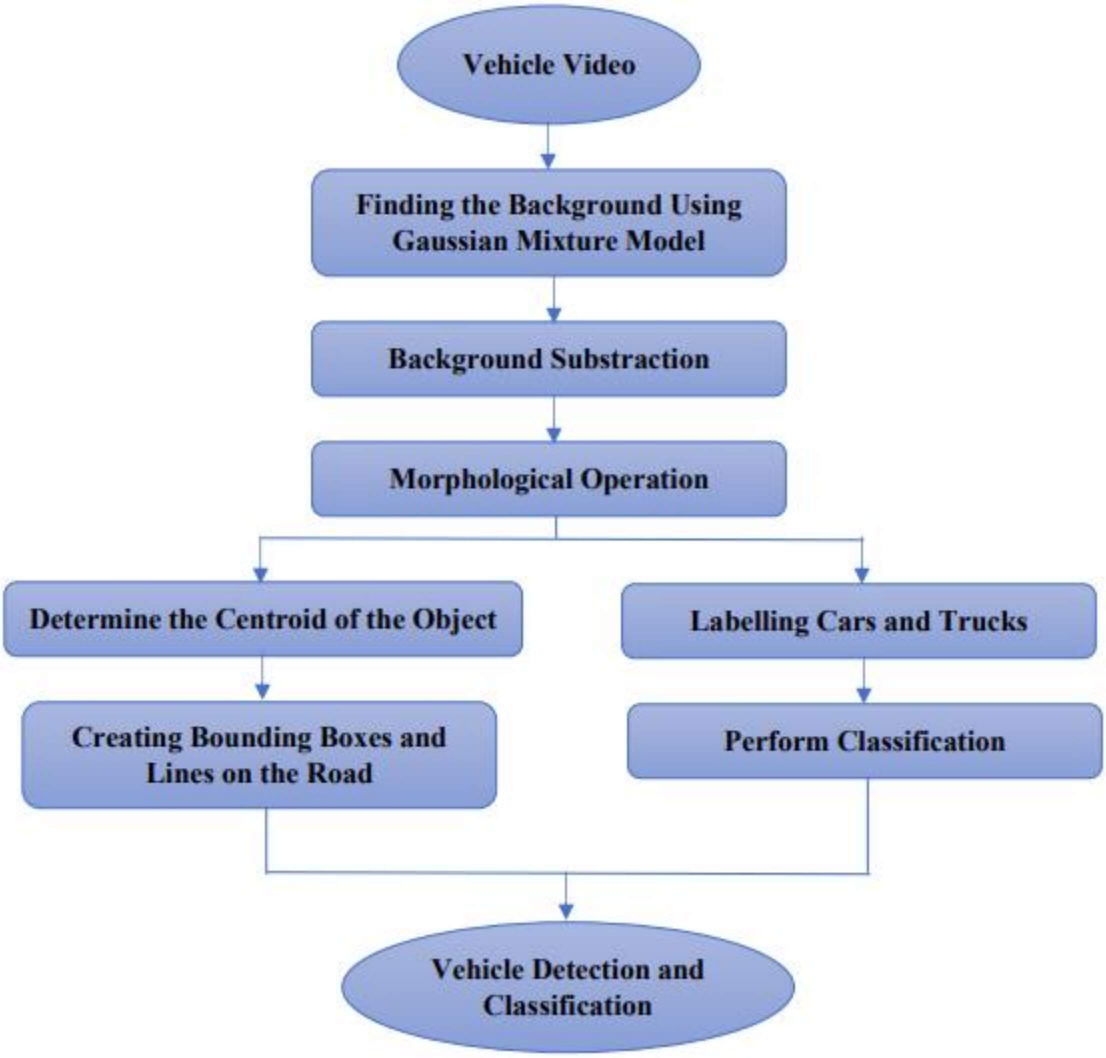


Figure 1. General Framework of This Research

The first stage in this research is taking video of the vehicle. Video shooting was carried out on a road in the Bintara Bekasi area. After that, to detect the vehicle, a video background search was carried out using the Gaussian Mixture Model method.

1. Gaussian Mixture Model

The first step in finding moving objects in a video is getting the background of the video. The background is an image in which there are objects that are not moving while an image that contains moving objects is called the foreground. In a video, the background and foreground are always changing, so we always have to update the background on each frame (Rostianingsih et al., 2008). Changes that may occur in the background are changes in light intensity from daytime to evening, changes in the shadows of objects in the background caused by changes in the position of the sun, changes in the position of objects in the background, adding objects in the background, and so on. To reduce errors that occur due to background changes, an adaptive background is needed, which is a background that can adjust to changes that occur in the background (Wang et al., 2013).

A common adaptive background method is averaging the combined pixel values of all consecutive inputs (Stauffer & Grimson, 1999). However, this method does not work well if given the input which has many moving objects. Therefore, in order to cover the shortage, in this study used methods of adaptive background mixture models such as Gaussian Mixture Models.

Gaussian Mixture Model (GMM) is a density model that consists of components of Gaussian functions (Gu et al., 2007). The components of this function consist of different weights to produce a multi-model density. In our research, GMM is used to model the background colors of each pixel. Each pixel has its own GMM and the data processed is the pixel color obtained from the input. GMM models are formed from time-based pixel color data. The model that is formed is divided into 2 parts, the background model and the foreground model.

The number of GMM models used will affects the number of background models. If the number of GMM models used is getting bigger then the number of background models owned by a pixel will increase. There are several stages in the GMM method, namely the stage of matching input to distribution and the stage of selecting a distribution that reflects the background. At this stage the inputs are matched against all distributions until the most suitable distribution is found. A pixel is said to be included in a distribution if the pixel value is within 2.5 standard deviation of a distribution as in equation (1).

$$\mu_k - 2.5\sigma_k < X_t < \mu_k + 2.5\sigma_k \,. \tag{1}$$

Where X_t is the vector of the pixel color (R, G, B) at time t, μ_k is the vector mean (R, G, B) from Gaussian to kth, and σ_k as the standard deviation from Gaussian to kth (Nascimento & Marques, 2004). If the pixel does not match all the existing distributions, then the pixel is considered as the foreground and a new distribution is created by replacing the distribution that at least reflects the background. The new distribution has mean values corresponding to pixel values, high variance values, and small weight values.

At this stage, an update is made on the value of the later GMM parameters used to process the next input. The updated values consist of weight, mean, and variance. Weight values are updated every time. To update the weight value, we use equation (2):

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{2}$$

$$M_{k,t} = \begin{cases} 0, & if\ the\ model\ not\ matched \\ 1, & if\ the\ model\ matched \end{cases} \,.$$

Where $\omega_{k,t}$ is the weight from Gaussian to kth at time t, α is the learning rate and the value of $M_{k,t}$ is 1 for the suitable model and 0 for the other models.

After the weight value is updated, normalization is carried out so that the total weight of all distributions is not more than 1. The mean value of a distribution is updated whenever a pixel value matches that distribution. To update the mean value, equation (3) is used:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{3}$$
$$\rho = \alpha\gamma(X_t|\mu_k, \Sigma_k)$$
$$\gamma(X_t|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)} .$$

Where Σ is the covariance matrix, $|\Sigma|$ is the determinant of the covariance, the power T is the transpose matrix, the power -1 is the inverse matrix, e is the exponent, π is phi, and n is the size of the vector X (R, G, B), where the covariance is obtained from equation (4):

$$\Sigma_{k,t} = \sigma_k^2 I . \tag{4}$$

With I is the identity matrix and σ_k^2 is the variance from Gaussian to kth. The standard deviation value of a distribution is updated whenever a pixel value matches that distribution. To update the standard deviation value, the formula (5) is used:

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) . \tag{5}$$

The next stage is choose background distribution. At this stage, models that reflect the background are selected. First, the models are sorted according to $\frac{\omega}{\sigma^2}$ so that the distribution that best reflects the background remains on top and that does not reflect the background at the bottom which will be replaced by another distribution. To select the first distribution which is used as the background distribution, the formula (6) is used:

$$\arg\min_b\left(\sum_{k=1}^{b} W_k > T\right) . \tag{6}$$

Where T is the smallest proportion of data that should be calculated as background. In this research, we use T = 0,7.

2. **Background Substraction**

Background substraction or also known as foreground detection is one of the steps in computer vision which is a process to detect significant changes that occur in a video frame (Putri, 2016). The purpose of background substraction is to separate the object and the background so that movement of an object can be detected. The result of background substraction is usually input which will be processed again at a more advanced level such as tracking the identified object.

The way the background substraction works is the frame when $\left(k+1, f(k+1)\right)$ is reduced by the frame at $\left(k, f(k)\right)$ the frame when k is considered the base frame or fixed frame, so that the difference in image position at $k+1$ (represented by the values of each pixel in the matrix) will be subtracted by the position of the image at time k (represented by the values of each pixel in the matrix). The image of the result of the background substraction process is shown in Figure 2.

3. **Morphological Operation**

Morphological operations are operations used in binary or black and white images to change the structure of the segment contained in the image (Kosasih, 2020). Segmentation is done by distinguishing between objects and backgrounds.

Morphological operations have 2 basic operations, namely dilation and erosion, and 2 other operations that are useful in image processing, namely the opening and closing which are formed from 2 basic operations of dilation and erosion (Susanto, 2019) .

Dilation is an operation on morphological operations that will increase the size of the object by changing the value of each pixel with the maximum value of the surrounding pixels. This can increase the number of each object's edge pixel in the image (Kadir, 2013). Dilation can be expressed by a formula (7).

$$g(x, y) = f(x, y) \oplus SE \ . \tag{7}$$

The erosion is the opposite of the dilation operation, in that the erosion operation will reduce the number of pixels by changing the value of each pixel with the minimum value of the surrounding pixels which will result in the loss of the object edge in the image (Umam & Negara, 2016). Erosion can be expressed by mathematical formula (8).

$$g(x, y) = f(x, y) \ominus SE \ . \tag{8}$$

The opening operation is a combination of the 2 basic operations of dilation and erosion. The step is to do erosion on the original image and then proceed with the dilation operation. This operation is used to remove small objects and smooth the boundaries of large objects without changing the object area significantly. The opening operation can be stated by the formula (9).

$$f(x, y) \circ SE = (f(x, y) \ominus SE) \oplus SE \ . \tag{9}$$

The closing operation is the opposite of the opening operation, where the step is to do the dilation first and then follow the erosion operation. The operation functions to remove small holes in the object, thereby combining adjacent objects and smoothing the boundaries of large objects without changing the object significantly. The closing operation can be stated by the formula (10).

$$f(x, y) \bullet SE = (f(x, y) \oplus SE) \ominus SE \ . \tag{10}$$

4. Bounding Box, Centroid and Boundary Line

Bounding box is a list that contains points (points) that represent a curve of an image. This picture can vary depending on the situation at hand. The bounding box is taken from the coordinates (x, y) of the object and w, h are the width and length of the bounding box itself. The purpose of providing bounding boxes is to mark, analyze, and recognize objects.

The bounding box uses the countour of the image that has been segmented using the previous morphological operation and then the object recognition process is carried out and the calculation of the curve points on the moving object. Then from these results the vehicles can be calculated and grouped based on the area of the bounding box.

To count the vehicles, we make a boundary line and centroid of vehicles. When the detected object has a centroid and the bounding box crosses the boundary line, the object in the form of a vehicle will be counted. The centroid is the equilibrium point of an object which can be determined by the formula (11) (Kosasih & Arfiansyah, 2020).

$$\bar{x} = \frac{\sum_{i=1}^{n} A_i x_i}{\sum_{i=1}^{n} A_i} \text{ dan } \bar{y} = \frac{\sum_{i=1}^{n} A_i y_i}{\sum_{i=1}^{n} A_i}. \tag{11}$$

Where A_i is the area of the ith object, x_i and y_i are the coordinates of the center of the object in each part, $\bar{x}$ and $\bar{y}$ are the coordinates of the center of the whole object.

The line is a collection of points that are infinite in number. If there are two points, namely (x_1, y_1) and (x_2, y_2) then a line can pass this points have equation which can be seen in equation (12) (Kosasih & Arfiansyah, 2020).

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1}. \quad (12)$$

5. **Classification**

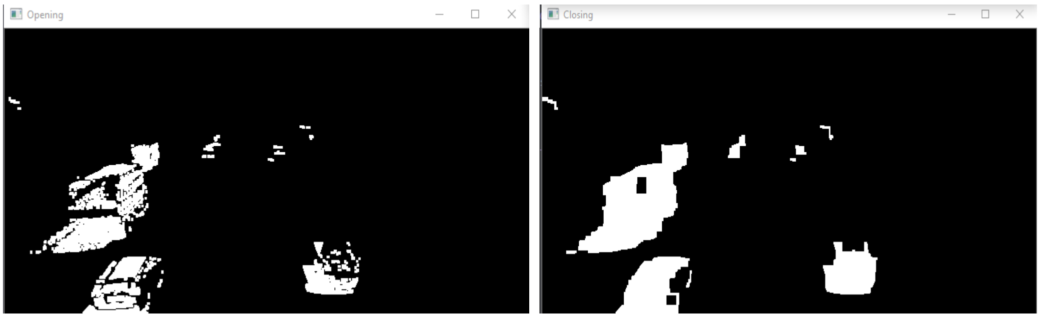
In this study, the detected vehicles will be classified into two types, namely cars and trucks. In vehicle classification, vehicle labeling is carried out using the area of the vehicle based on the resulting morphological image.

RESULTS AND DISCUSSION

Based on the research stages that have been carried out, the data used is vehicle video data in the Bintara area, Bekasi. To detect a vehicle, the first stage is to look for the background of the video. Gaussian Mixture Models method is used to find the background of the video. After getting the background, then do the background substraction, which is to find the difference between the original frame in the video and the background of video. The result of the background substraction is a foreground image which can be seen in Figure 2.

Figure 2. Result of Background Substraction (a. Foreground Image, b. Background Image)

Based on Figure 2, it can be seen that there are unwanted objects that appear in the foreground image, so the image is done repair. In this study, morphological operations were used to improve the foreground image as shown in Figure 3.



(a) (b)

Figure 3. Morphological Operation (a. Opening Operation, b. Closing Operation)

After obtaining a moving object (vehicle), the next step is to make a bounding box based on the countur of the morphological operation. In this stage, the centroid of each moving object is also made which is used to calculate the number of vehicles. Because there are so many moving objects, in this study it is limited by making a straight line, so that only vehicles that are detected are passing the line as shown in Figure 4.

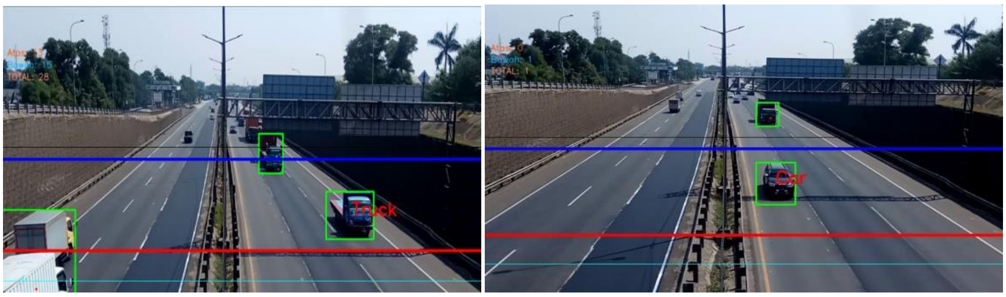


Figure 4. Example of Bounding Box

In this study, (Kosasih, 2021; Lestari et al., 2019) to evaluate the performance of our model we use indicators such as precision, recall and accuracy which can be seen in equations (13), (14) and (15).

$$Precision = \frac{TP}{(TP+FP)}\ x\ 100\% \tag{13}$$

$$Recall = \frac{TP}{(TP+FN)}\ x\ 100\% \tag{14}$$

$$Accuracy = \frac{(TN+TP)}{(TN+TP+FN+FP)}\ x\ 100\%\,. \tag{15}$$

Where: TP (True Positive) is a vehicle detected in real conditions and declared as a vehicle in the application, FP (False Positive) is no vehicle detected in real conditions but declared a vehicle in the application, FN (False Negative) is a vehicle detected in that condition real but not stated in the application and TN (True Negative) is no vehicle detected in real conditions and not stated in the application.

In the test results in the vehicle video, for (FN) vehicles detected in real conditions but not stated in the application, 2 vehicles were obtained as in Figure 5.

Figure 5. Example of False Negative (FN)

Furthermore, (FP) no vehicle is detected in real conditions but otherwise the vehicle in the application obtained by 7 vehicle as shown in Figure 6.

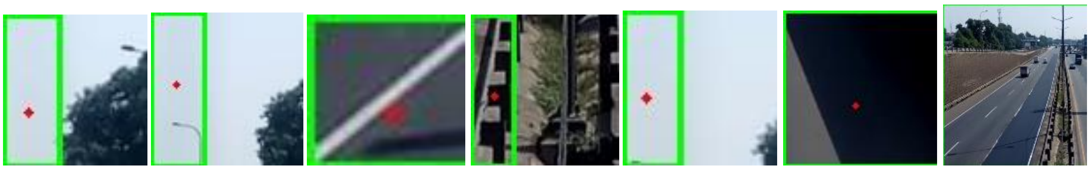


Figure 6. Example of False Positive (FP)

Next, TP is a vehicle that is detected in real conditions and is declared as a vehicle in the application. From this result, there are 57 vehicles as shown in Figure 7.

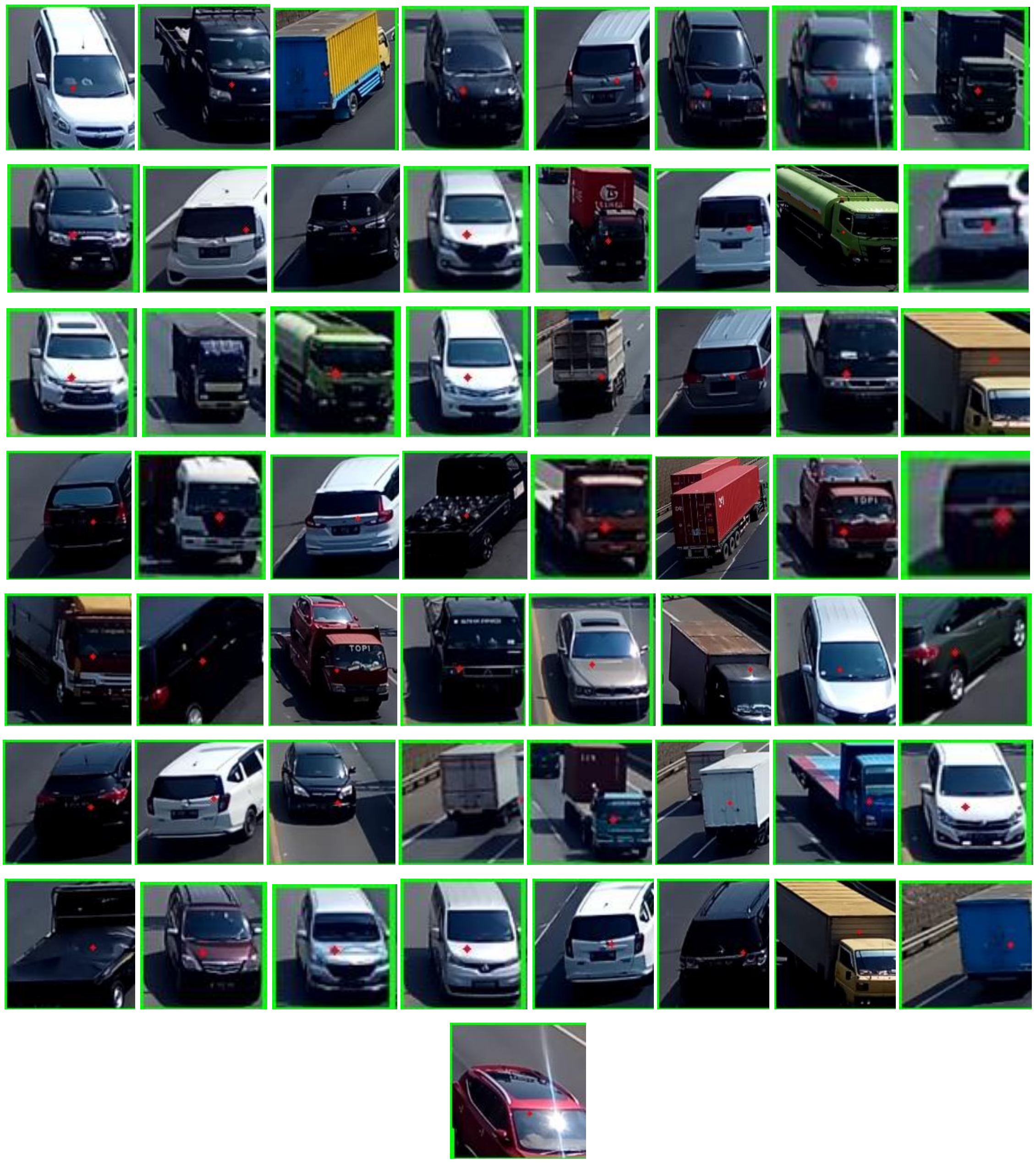


Figure 7. Example of True Positive (TP)

Based on Figure 5, 6 and 7, we calculate precision, recall and accuracy by using equation (13), (14) and (15) which can be seen in Table 1.

Table 1. Evaluation Model

Indicator	*Result*
True Positive (TP)	57
True Negative (TN)	0
False Positive (FP)	7
False Negative (FN)	2
Precision	89.0625%
Recall	96.6101%
Accuracy	86.3636%

Based on Table 1, it can be seen that the precision value is 89.0625%, the recall value is 96.6101% and the accuracy value is 86.3636%. The next stage is to classify the vehicle on the vehicle detection results.

Based on Figure 7, there are 57 vehicles detected in real conditions and declared as vehicles in the application. Furthermore, the classification of 57 vehicles was carried out using the vehicle area. The detected vehicles are classified into 2 types, namely cars and trucks. The classification results can be seen in Figure 8.

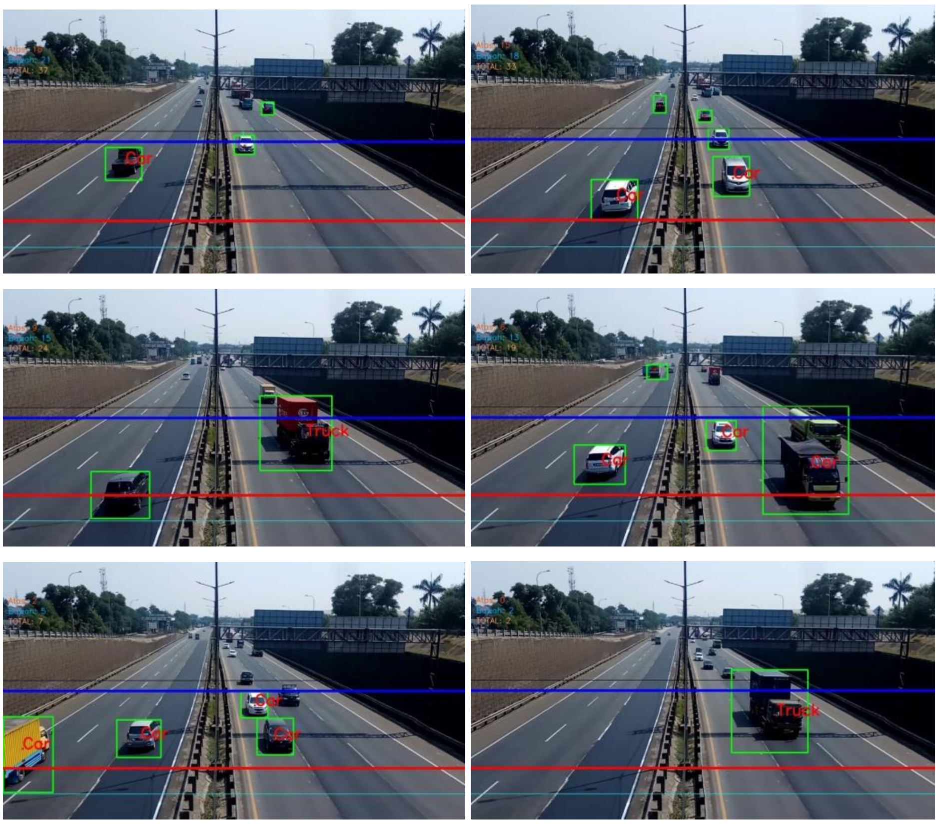


Figure 8. Frame results from vehicle classification

Based on the result frame from the classification, we perform evaluation by using a confusion matrix which can be seen in Table 2.

Table 2. The Result of Classification

	Predict Car	*Predict Truct*
Car	35	2
Truck	6	14

From Table 2, it can be seen that there are 35 cars in real conditions predicted by the application, the number of cars in real conditions is predicted by trucks there are 2. Trucks in real conditions are predicted by the application there are 6, the number of trucks in real conditions predicted by trucks there are 14. Furthermore, the calculation of the level of accuracy is carried out based on Table 2.

$$\text{Classification accuracy} = \frac{35+14}{35+14+6+2} \times 100\% = \frac{49}{57} \times 100\% = 85.9649\%$$

CONCLUSIONS AND RECOMMENDATIONS

Traffic monitoring is a very important thing to do to know the traffic condition. The security forces usually supervise via CCTV or come directly to the location they want to be watched. However, this surveillance is not possible for a full day, so a traffic control method is needed, such as detecting a vehicle without having to monitor it all day long. In this study we use the Gaussian Mixture Model method and morphological operations to detect vehicles. The data used in this study is traffic video data in the Bintara Bekasi area.

After the video data is obtained, the next step is to find the background of each frame in the video using the Gaussian Mixture Models method and perform background substraction to get an image of a moving object (vehicle). The image resulting from the background substraction is also called a foreground image. The next step is to repair the foreground image using image morphology. The results of morphological images are used to detect vehicles by making a bounding box on the vehicle. From the research results, it was found that 57 vehicles were detected from 66 vehicles passing through the area, so an accuracy rate of 80.43% was obtained.

The next stage is to classify the vehicles into 2 types, namely cars and trucks. Classification of vehicles is based on the area of the vehicle. Based on the classification results of 57 detected vehicles, it is found that there are 35 cars in real conditions predicted as cars, 2 cars in real condition predicted trucks, 6 trucks in real conditions predicted by the application and there are 14 trucks in real conditions predicted as trucks. From these results obtained a classification accuracy rate of 85.9649%.

For further research, to classify vehicles, other methods are used as a comparison, such as deep learning methods and others.

REFERENCE

Gu, J., Chen, J., Zhou, Q., & Zhang, H. (2007). Gaussian Mixture Model of Texture for Extracting Residential Area from High-Resolution Remotely Sensed Imagery. *The 5th ISPRS Workshop on DMGISs*, 157–162.

Kadir, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Andi.

Kosasih, R. (2020). Automatic Segmentation of Abdominal Aortic Aneurism (AAA) By Using Active Contour Models. *Scientific Journal of Informatics*, *7*(1), 66–74. doi: 10.15294/sji.v7i1.23625

Kosasih, R. (2021). Pendeteksian Kendaraan Menggunakan Metode Median Filter. *Jurnal Ilmiah Komputasi*, *20*(1), 53–58. doi: 10.32409/jikstik.20.1.402

Kosasih, R., & Arfiansyah, M. (2020). Pendeteksian Kendaraan dengan Menggunakan Metode Running Average Background Substraction dan Morfologi Citra. *Jurnal Media Informatika Budidarma*, *4*(4), 979–985. doi: 10.30865/mib.v4i4.2315

Kosasih, R., Fahrurozi, A., & Mardhiyah, I. (2020). Vehicle Detection Using Principal Component Analysis. *Jurnal Ilmiah Komputasi*, *19*(2), 155–160. doi: 10.32409/jikstik.19.2.83

Lestari, D. P., Kosasih, R., Handhika, T., Murni, Sari, I., & Fahrurozi, A. (2019). Fire Hotspots Detection System on CCTV Videos Using You only Look Once (YOLO) Method and Tiny YOLO Model for High Buildings Evacuation. *2nd International Conference of Computer and Informatics Engineering IC2IE*, 87–92. doi: 10.1109/IC2IE47452.2019.8940842

Nascimento, J. C., & Marques, J. S. (2004). Novel Metrics for Performance Evaluation of Object Detection Algorithms. *6th International Workshop on Performance Evaluation for Track- Ing and Surveillance*, 7–14. doi: 10.1109/TMM.2006.876287

Putri, N. N. (2016). Aplikasi pendeteksi objek bergerak pada Image Sequence Dengan Metode Background Substraction. *Jurnal IlmiahTeknologi Dan Rekayasa*, *21*(3), 162–172.

Rad, A. G., Dehghani, A., & Karim, M. R. (2010). Vehicle Speed Detection in Video Image Sequences Using CVS Method. *International Journal of Physical Sciences*, *5*(17), 2555–2563.

Rostianingsih, S., Adipranata, R., & Wibisono, F. S. (2008). Adaptive Background Dengan Metode Gaussian Mixture Models Untuk Real-Time Tracking. *Jurnal Informatika*, *9*(1), 68–77. doi: 10.9744/informatika.9.1.68-77

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive Background Mixture Models for Real-Time Tracking.

In Computer Vision AndPattern Recognition.

Susanto, A. (2019). Penerapan Operasi Morfologi Matematika Citra Digital Untuk Ekstraksi Area Plat Nomor Kendaraan Bermotor. *Pseudocode*, *6*(1), 49–57. doi: 10.33369/pseudocode.6.1.49-57

Tian, Q., Zhang, L., Wei, Y., Zhao, W., & Fei, W. (2013). Vehicle Detection and Tracking at Night in Video Surveillance. *International Journal of Online Engineering*, *9*(6), 60–64. doi: 10.3991/ijoe.v9iS6.2828.

Umam, K., & Negara, B. S. (2016). Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Subtraction Dan Operasi Morfologi. *CoreIT*, *2*(2), 31–40. doi: 10.24014/coreit.v2i2.2391

Wang, X., Sun, J., & Peng, H. (2013). Foreground Object Detecting Algorithm based on Mixture of Gaussian and Kalman Filter in Video Surveillance. *Journal of Computers*, *8*(3), 693–700. https://doi.org/10.4304/jcp.8.3.693-700. doi:10.4304/jcp.8.3.693-700