http://ejournal.seaninstitute.or.id/index.php/InfoSains
Jurnal Info Sains : Informatika dan Sains, Volume 13, No 02 2023
E-ISSN.2797-7889, P-ISSN.2089-3329

Implementation of Object Tracking to Detect and Count the Number of Vehicles Automatically Using the Kalman Filter and Gaussian Mixture Model Methods

[1]Marsiska Ariesta Putri, [2]Martinus Apun Heses, [3]Kristiawan Nurdianto
[1,2,3] Institut Teknologi Dan Bisnis Semarang
Email: siskaloyal99@gmail.com

Keywords

Vehicle counting system, Object Tracking, .Gaussian Mixture Model, Kalman Filter.

Abstract. Traffic density can be controlled by obtaining and managing vehicle volume data on the road. In general, the process of acquiring data on the volume of vehicles passing on the highway is still carried out manually, namely by assigning several people to the field and counting each vehicle that passes, then dividing it by a certain time period. In manual calculations there are still many weaknesses such as data collection taking a long time, and the large amount of human resources required. Based on these conditions, an accurate automatic vehicle counting and detection system is needed as a traffic control monitor and traffic analyzer.Currently, a vehicle detection system has been developed using sensors, Radio Frequency Identifiers or other hardware which is integrated by software on a microcontroller and works automatically to detect speed and count the number of vehicles passing on the highway. The weakness of this detector is that it can only detect a narrow range, the system design and operations are complicated, and the operational costs are large. Based on these several things, this research was developed with a focus on designing a vehicle detection and counting system using the Kalman Filter and Gaussian Mixture Model (GMM) methods.This research intends to design and create a program that is able to identify the type of vehicle in a video input and calculate the number of vehicles detected based on their type. Users simply enter traffic recording videos into the Car Detection program. Later, the program will process the input video and produce a .txt file as output. This file contains the types of vehicles detected along with the number of vehicles detected based on their type. Using the Kalman Filter and Gaussian Mixture Model (GMM) methods, the most accurate results were obtained in the morning (lighting 10,000-25,000 lux) with an F1 Score of 0.91111, while the least accurate vehicle counts occurred in the evening (lighting 0.27 -1.0 lux) with an F1 Score of 0.16071

1. INTRODUCTION

Traffic density must always be monitored so that traffic infrastructure can be managed well, so that traffic congestion can be reduced. Vehicle volume data can be used for the process of calculating vehicle density, calculating vehicle frequency, predicting congestion, and can also be used as a reference for repairing existing roads, widening roads, adding new roads, creating or arranging new routes, arranging traffic signs and improving transportation infrastructure others. Traffic conditions on the road have begun to be monitored using CCTV. This CCTV will later produce recordings of traffic conditions. This recording can be used for various purposes. One of them is to detect the type of vehicle passing through the road.In general, the process of acquiring data on the volume of vehicles passing on the highway is still carried out manually, namely by assigning several people to the field and counting each vehicle that passes, then dividing it by a certain time period. In manual calculations there are still many weaknesses such as data collection taking a long time, and the large amount of human resources required.

The program will recognize vehicle objects contained in the recorded video. Recognized vehicle objects can be classified into several types. For example, the detected vehicle objects are classified into three types, namely small cars, medium cars and large cars. Later, a count can be made of each type of vehicle that passes the road. So you can know the number of cars based on their type that use that road. Currently, a vehicle detection system has been developed using sensors, Radio Frequency Identifiers or other hardware which is integrated by software on a microcontroller and works automatically to detect speed and count the number of vehicles passing on the highway. The

weakness of this detector is that it can only detect a narrow range, the system design and operations are complicated, and the operational costs are large.In the field of computer vision, a visual automatic vehicle detection system has been developed which is used to detect and calculate dynamically moving vehicles efficiently. Intelligent visual monitoring of vehicles on the road is an important component for developing intelligent transportation systems. Based on these several things, this research was developed with a focus on designing a vehicle detection and counting system using the Kalmanfilter and Gaussianmixture model (GMM) methods.

2. METHOD

System data design process

In the first step, a highway survey is carried out with moderate or not too dense traffic flow to determine the place and time for data acquisition. The highway must be crossed by a pedestrian bridge or similar to place the webcam and tripod. The angle required for image acquisition is 65o - 90o from a flat road to get a good image angle. The camera is set in such a way that it does not move/is always in a stable condition so that acquisition results are good. Data acquisition was carried out at 4 different times of morning, afternoon, evening and night to determine the effect of lighting on processing results.

After determining the place and time for acquisition, the light intensity values are then measured at any time (morning, afternoon, evening and night) followed by data acquisition. Acquisition is carried out in the form of image recording using a webcam. A webcam connected to a PC will save video files resulting from image acquisition. Then the image acquisition results are processed using the image segmentation method with the GMM (Gaussian mixture model) function to detect background and foreground images. The background image is identified with a still part of the video, while the foreground image is a moving object in the video. Creating a detecting lane and initializing the background image so that it is divided into 2 areas, areas A and B.

The Kalman filter function is provided to predict frameforeground that has been given an ID by GMM. When one ID moves, the Kalman filter will predict the movement of the ID for each frame from area A to area B or vice versa, because the initialized background can be used in both directions (from A to B or B to A). The ID frame foreground filtered by the Kalman filter and GMM is given an ROI (Region of Interest) in the form of a rectangle to limit the ID frame pixels and as a tracking visualization. If the ID frame tracked from the initial area has passed the detecting lane, then the ID will be counted one by one. automatically because the ID is already in the next area. Then the system will update tracking the new vehicle ID to continue the iteration, so that the number of vehicles (ID) will be accumulated and displayed on the video.

Framework

System flowchart

The system works automatically starting from image segmentation, identifying image background-foreground, tracking, counting, to displaying the results of image processing. The results displayed are in the form of data processing videos and binary videos to show the Kalman and Gaussian functions.

System Performance Testing

In performance testing, the system will read the video frame by frame. This reading is carried out in a previously determined video area, namely the area outlined in blue. Outside this area, vehicles will not be detected. This is done to reduce the risk of detection errors. Thus, vehicle object recognition will become more focused. So that the recognition of the type of vehicle detected will be more accurate.

Then proceed with detecting vehicle objects using a classifier. The classifier will match the pre-loaded XML file vehicle objects. If it meets the requirements, the symmetry of the detected vehicle object will then be checked. The detected vehicle object will be checked whether it is symmetrical vertically and horizontally or not. If it is symmetrical vertically and horizontally, the object is

considered a vehicle. The application will provide a rectangular mark on objects that are considered to be vehicles.

This system obtains qualitative F1 Score results with the relationship between manual counting (CM) and automatic counting (CO) with truepositive (TP), falsepositive (FP) and falsenegative (FN) parameters. FP and Fncounting are the error values of the system. The F1 Score value ranges from 0 to 1, the closer the value is to 1, the more accurate it is. F1 Score is the result of an accuracy test. TP (True positive) is a target pixel that is recognized as an object. FP (False positive) is noise that is detected as an object. FP (False negative) is a target that is not detected as an object.

Validation and evaluation

Mean squared error is used to determine the difference between the estimator and the estimation results. MSE is used as a benchmark for an estimator. The results obtained are always positive numbers. The closer to zero, the better the estimator's performance. In this research, MSE is used to check whether an image is a vehicle object or not. This is done with Compare the horizontal and vertical symmetry of an image. A vehicle image is symmetrical horizontally and vertically. Horizontal symmetry is used to determine the height of the vehicle object, while vertical symmetry is used to determine whether the objects detected on the left and right sides are symmetrical or not. This can make it easier to mark detected vehicle objects. If the symmetry difference is too large, the detected image is not a vehicle object.

3. RESULTS AND DISCUSSION

The topics that At this stage, the steps taken in the system are explained in stages, starting from the feature extraction process, recognizing the type of vehicle in the video, until it becomes the output result.

Object Detection Processing

At this stage, the steps taken in the system are explained in stages, starting from the feature extraction process, recognizing the type of vehicle in the video, until the output results are obtained. First, the XML file contains features that represent a vehicle object. Each candidate vehicle object is matched to the XML file. If it meets the specified criteria, the object will be considered a vehicle object. And the object is marked with a red rectangle.This is done to reduce the risk of detection errors. Thus, vehicle object recognition will become more focused. So that the recognition of the type of vehicle detected will be more accurate. Then proceed with detecting vehicle objects using a classifier. The classifier will match the pre-loaded XML file vehicle objects. If it meets the requirements, the symmetry of the detected vehicle object will then be checked. The detected vehicle object will be checked whether it is symmetrical vertically and horizontally or not. If it is symmetrical vertically and horizontally, the object is considered a vehicle. The application will provide a rectangular mark on objects that are considered to be vehicles

Conducting Classifier Training

Recognition of vehicle types is done by measuring the area of the rectangle used to mark the vehicle object. After identifying the vehicle type, the application will record the vehicle ID and type in the .txt file. This is to make it easier for users to find out the type of vehicle that has been detected from the video entered.

Following are the steps taken.

Point opencv_traincascade to positive samples (samples.vec), negative images. Write the results into our repository clustering directory and sample sizes (-w and -h). The variable -numNeg determines how many negative samples there are. The precalcValBufSize variable and the -precalcIdxBufSize variable determine how much memory will be used during training. Please remember, the variable -numPos must be lower than the positive samples generated. After starting the training process, the process will print the parameters again and then start the training process. Each

stage will print some analysis. At the end of each stage, the classifier is saved to a file and the process can be stopped and restarted (if the device is turned off). When the process is complete we will find a file called classifier.xml in the classifier directory.

Object Detection Region Implementation

This process is carried out to determine the detection area for vehicle objects in the video. In this process, the vehicle object is reduced and the boundaries of the vehicle object detection area in the video are determined. Every time the classifier detects a vehicle object, the object will be marked with a maximum time limit with an initial value of 0 (located fourth in the region list, explained further in the next subchapter). This value will increase by 1 as long as the vehicle object enters the detection area. This value is also useful as a parameter to determine the speed of vehicle objects. The aim is to mark vehicle objects detected in the ROI area. If the distance between a vehicle object and another vehicle object is less than 75 pixels, the two objects are considered as one vehicle object. to read the input video frame-by-frame.

The following source code also includes delineating the vehicle object detection area. So the classifier only detects vehicle objects in that area.

Functionality Testing Accuracy

In testing, there were There are three road conditions, namely quiet, congested and very congested. These three conditions represent real conditions that occur on highways in general. The accuracy of vehicle type recognition is calculated using the equation below which is calculated in the following way:

$$Acc=\square(V\_detect/V\_real\ )+100\%$$

Where Vdetect is the number of a type of vehicle detected and Vreal is the actual number of a type of vehicle. Vreal is obtained by manually calculating each type of vehicle.

The following are the test results obtained for quiet road conditions. In this test, the author used 4 maximum time limit values, namely 25, 30 and 35. If the time required for an object to pass through the detection area exceeds these values, the object will be detected as a vehicle more than once. Thus, vehicle counting will be inaccurate. The results obtained were calculated for accuracy using Equation 9. The original numbers of small, medium and large vehicles were respectively 1, 3 and 1. In this test, there was a detection error, namely that the medium vehicle was detected as a large vehicle. This is caused by the wrong angle of view of the vehicle object towards the camera. So the object looks bigger than its actual size. For quiet road conditions, the average accuracy is 77.8% and the ideal maximum time limit value is 35. In this test, the author used 4 maximum time limit values, namely 25, 30 and 35. If the time required for an object to pass through the detection area exceeds these values, the object will be detected as a vehicle more than once. Thus, vehicle counting will be inaccurate. Thus, vehicle counting will be inaccurate.

From the data, accuracy was calculated using Equation 9. The original number of small, medium, and large vehicles were respectively 10, 8, and 4. Detection accuracy for small, medium, and large vehicles at the maximum time limit of 25 and 30 was 30%, 25 %, and 75%. The average accuracy for maximum timeouts 25 and 30 was 43.3%. There is a detection error at the maximum time limits of 25 and 30, namely the vehicle is being detected as a large vehicle. This is because the vehicle object moves slower than the maximum time limit value set. And also, the distance between vehicle objects also influences detection accuracy. In this test, the distance between vehicles tends to be closer than in tests with quiet road conditions. For maximum time limits of 35 and 40, the results obtained for small, medium and large vehicles are 25%, 37.5% and 75% respectively. No errors were found in the classification of vehicle types at the maximum time limit of 35. For normal road conditions, the limit value The ideal maximum time is 35 with an average accuracy of 47.5%.

Test Results for Congested Road Conditions

Test results obtained for congested road conditions. In this test, the author used 4 maximum time limit values, namely 25, 30 and 35. If the time required for an object to pass through the detection area exceeds these values, the object will be detected as a vehicle more than once. Thus, vehicle counting will be inaccurate.From the data, accuracy calculations were carried out using Equation 9. The original number of small, medium, and large vehicles were respectively 9, 8, and 8. Detection accuracy for small, medium, and large vehicles at the maximum time limits of 25 and 30 was 11.1% , 25%, and 37.5%. The average accuracy for maximum timeouts 25 and 30 was 24.5%. For a maximum time limit of 35, the detection accuracy of small, medium, and large vehicles is 22.2%, 25%, and 37.5%, respectively.

The average accuracy for a maximum timeout value of 35 was 28.2%.In this test, there was an error in classifying the type of vehicle, namely that the medium vehicle was detected as a large vehicle. There are also small vehicles that are detected as large vehicles. In this test, these two events were influenced by two factors. The first factor is the distance between vehicle objects. In this test, the distance between vehicles tends to be closer than in tests with quiet road conditions. Then, the second factor is the camera angle. In this test, there are small cars that look like big cars. This is because the camera highlights the object from the side (not right from the middle). As a result, objects appear larger than their actual size. For busy road conditions, the ideal maximum time limit value is 35 with an average accuracy of 28.2%.

4. CONCLUSION

From the results of observations during the design, implementation and software testing process, the following conclusions can be drawn, vehicle types can be classified based on the rectangular area of the vehicle object marker. The optimal maximum time limit for each vehicle object in passing through the detection area is 35 milliseconds . Thus reducing the risk of the same vehicle object being detected more than once. The average level of accuracy for three different road conditions (quiet, normal, congested) is 77.8%, 47.5%, and 28.2%. These suggestions are based on the design results, The implementation and testing that has been carried out enriches the features of other vehicle types in the XML file which is used as a template for detecting vehicle objects. So the program can still recognize new types of vehicle objects in the future. Adding a camera from another point of view and another algorithm. So that detection accuracy can increase. Added a feature to connect the program with the camera. So the program is able to detect vehicle types in real-time.

REFERENCES

[1] Intel Corporation, "OpenCV," Itseez, June 2000. [Online]. Available: http://opencv.org/. [Accessed 4 December 2016].
[2] Intel Corporation, "OpenCV," Itseez, June 2000. [Online]. Available: http://opencv.org/platforms/. [Accessed 5 December 2016].
[3] P. Viola and M. Jones, "Rapid Object Detection using A Boosted Cascade of Simple Features," in Conference on Computer Vision and Pattern Recognition CVPR 2001, Hawaii, 2001.
[4] P. Viola and M. J. Jones, "Robust Real-time Face Detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004.
[5] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," in International Conference on Image Processing 2002, New York, 2002.
[6] Office for Mathematics, Science, and Technology Education University of Illinois, "Mean Square Error," Office for Mathematics, Science, and Technology Education (MSTE), Champaign, 2004.
[7] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," International Journal of Computer Vision, vol. 38, no. 1, pp. 15-33, 2000.

[8] V. K. Narayanan, "Vision Based Robust Vehicle Detection and Tracking VIA Active Learning," University of Florida, Gainesville, 2013.
[9] Badan Pengembangan dan Pembinaan Bahasa, "KBBI Daring," Kementerian Pendidikan dan Kebudayaan
Indonesia, 2016. [Online]. Available: https://kbbi.kemdikbud.go.id/entri/kendaraan. [Accessed 9 May 2017].
[10] M. Oliveira and V. Santos, "Automatic Detection of Cars in Real Roads using Haar-like Features," Department of Mechanical Engineering, University of Aveiro, vol. 3810, 2008.
[11] S. Han, Y. Han and H. Hahn, "Vehicle Detection Method using Haar-like Feature on Real Time System," World Academy of Science, Engineering and Technology, vol. 59, pp. 455-459, 2009.
[12] S. Sivaraman and M. M. Trivedi, "A General Active- Learning Framework for on-road Vehicle Recognition and Tracking," IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 2, pp. 267-276, 2010.
[13] T. Ball, "Coding Robin," Coding Robin, 22 July 2013. [Online]. Available: http://coding- robin.de/2013/07/22/train-your-own-opencv-haar- classifier.html. [Accessed 23 May 2017].