2018 3rd International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia

Vehicle Tracking using Kalman Filter based on Smart Video Sensor Architecture

Imelda Imelda, IEEE Member
Department of Computer Science
Faculty of Information Technology
Universitas Budi Luhur
Jakarta, Indonesia
imelda@budiluhur.ac.id

Agus Harjoko
Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences Universitas Gadjah Mada
Yogyakarta, Indonesia
aharjoko@ugm.ac.id

Pekik Nurwantoro
Department of Physics
Faculty of Mathematics and Natural Sciences Universitas Gadjah Mada
Yogyakarta, Indonesia
pekik@ugm.ac.id

***Abstract*— Traffic information is needed to determine the cause of the accident. Problems arise when many traffic accidents or violations co-occur. Technical failures in delivering important frames also hinder the process of analyzing the video, which occurs due to disconnected network, limited bandwidth and CPU processing power. Besides, the size of the video to be processed at the same time slow the CPU down preventing the video from being treated. In this research, we propose Smart Video Sensor (SVS) resolve the missing frame issues. SVS is a video sensor recording images streaming frames for the frame. SVS extract only features of traffic objects and compress the video so that the data will be received faster and lighter. SVS also processes the primary data, so the other system is ready to use the features needed for further data processing. To demonstrate how well SVS works, we experimented it by tracking vehicles by type. This study uses 3 locations and 1000 frames in each area. The contribution of this paper is to produce a vehicle tracking model by type using Kalman Filter based SVS Architecture. The highest accuracy found for motorcycles is in Galeria (90.71%).**

Keywords—Smart Video Sensor, vehicle tracking, features of traffic objects, Kalman Filter

I. INTRODUCTION (*HEADING 1*)

In general, traffic issues include traffic density, traffic violations and traffic accidents [1]. The Traffic Monitoring and Management System is needed by the police so that traffic issues can be appropriately handled. Fig.1. shows a recent Yogyakarta Traffic Monitoring and Management System. At the Regional Traffic Management Center (RTMC), the Police can monitor traffic from various CCTV owned by the Department of Transportation (Dishub), the Communication and Informatics Service (Diskominfo) and the Traffic Service (Dislantas) because the data is integrated through the internet. In the RTMC Dislantas room, the Area Traffic Control System (ATCS) Dishub and JOGJA PROP Diskominfo, many video recordings are viewed concurrently. In these spaces, there are large monitors of several servers and several personal computers (PCs) / clients. CCTV in real time sends video data to the server.

Yogyakarta police still monitor traffic manually since 2007 [2] to this date. The manual monitor means that the operator directly controls the traffic from all Closed Circuit Television (CCTV) in Yogyakarta. Sometimes the police need video data from CCTV owned by Dishub and Diskominfo Yogyakarta for evidence of traffic violations, court evidence or proof of accident. If the police need it, they must ask the Dishub and Diskominfo according to the policy procedures set by the management of Dishub and Diskominfo because the right of ownership of the video data belongs to each relevant office and data is confidential. Dishub and Diskominfo will check the availability of video data because the video recording period is only 30 days.

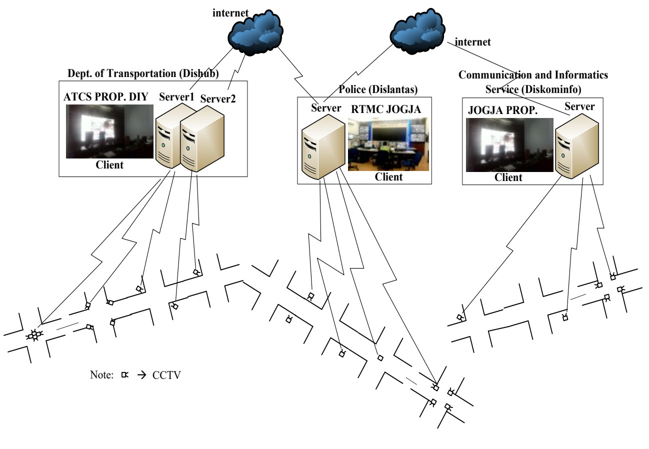

Fig. 1. A recent Yogyakarta Traffic Monitoring and Management System

In 2012, the Transportation Department stated that the problem faced was that video data was often received fragmented because many video frames were lost. As a result, video data cannot be processed further. Failure to present these crucial frames inhibits the video analysis process. This failure also impedes the process of obtaining traffic information regarding the latest traffic conditions, causes of accidents and traffic problems that often occur together in several different places. Authentic evidence in the form of video data cannot be obtained maximally for proceedings in the court if the relevant Police and Service cannot utilize video data to the fullest because video data cannot be processed further. Video data is needed to get accurate information about the overall picture at the crime scene. For example, the speed of the vehicle just before and after an accident. This process is an audit process that requires vehicle tracking.

The solution to the problem is to modify the concept of the Smart Video Sensor Architecture (SVS) from research conducted initially by [3]. The proposed SVS is a video sensor that extracts features of traffic objects and compresses video data. SVS retrieves the characteristics of objects in memory and forms the features of objects in XML format. Fig.2. shows an illustration of the benefits of SVS on classified vehicle tracking. This illustration explains how simple features in XML format are sent from SVS to DCCU

 978-1-5386-7082-8/18/$31.00 ©2018 IEEE

(Data Center and Control Unit) or LPU (Local Processing Unit).

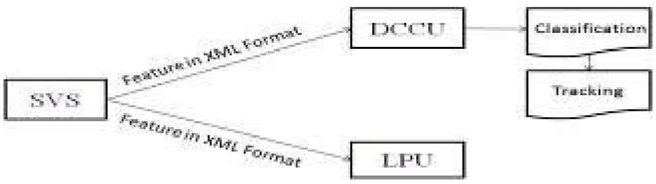

Fig. 2. Illustration of the benefits of SVS in classified vehicle tracking

SVS is placed at every intersection in Yogyakarta area in line with the number of CCTV cameras in the Yogyakarta area. This goal to overcome the problem of the number of video frames lost in each city that has CCTV. SVS does not store object characteristics in files. SVS immediately sends compressed video data and features of traffic objects in XML format to LPU and DCCU. Fig. 3 shows the proposed Yogyakarta Traffic Monitoring and Management System. SVS results are sent to the LPU for processing locally. SVS results also are posted to DCCU so that they can be used for further traffic data analysis. LPU is a processing unit for managing traffic locally that does not store SVS results. The LPU performs feature processing, and the results are directly used to monitor and regulate traffic lights. LPU is placed in every intersection of Yogyakarta that requires local traffic control. Every junction of the Yogyakarta area is quite an LPU because it only processes traffic arrangements locally. DCCU stores SVS results in the file structure, which contains data and applications stored on a server that is in an organization. Those applications made in DCCU are needed by the police, Dishub and Diskominfo Jogja. DCCU can control traffic from a distance and can track, classify and regulate traffic from the Yogyakarta Traffic Department Area Traffic Control System (ATCS), Jogja Prop. Diskominfo and RTMC Jogja. DCCU can be in the form of a dashboard that monitors all activities in each JOGJA ATCS, JOGJA RTMC, and JOGJA PROP. DCCU can be placed in every related Office. In this study, DCCU was set in JOGJA ATCS, JOGJA RTMC, and JOGJA PROP because each related Service requires controlling traffic from far away. This very useful if Green Wave, that occurs when a series of traffic lights (usually three or more) are coordinated to allow continuous traffic flow over several intersections in one main direction, is set in the DCCU, DCCU can arrange for vehicles that go through the big road to get priority green lights in sequence.

The applications in DCCU allows obtaining flow, speed, and density which are traffic parameters. The fundamental characteristics of traffic flow are flow, speed, and density [4]. Traffic parameters are a benchmark in the traffic transportation system [5]. Traffic parameters are characteristics of traffic flows that can be classified based on quantifying measures, quality assessment measures, movement measures and composition/classification measures. Parameter traffic is needed to help solve traffic problems. When traffic parameters are known, the density of vehicles can also be identified based on traffic flow and density of vehicle traffic. The possibility of an accident can be anticipated earlier through the track or path traversed by a vehicle and the speed of the car that exceeds the maximum speed standard somewhere. An indication of the possibility of an accident can be seen from the vehicle driving uncontrollably or moving from one track to another with irregularities, and the speed of the vehicle exceeds the authorized speed limit. If the LPU is placed at each intersection in sequence, the car can be tracked by calculating its speed. The existence of the vehicle is known by identifying vehicle features. The car features detected in an LPU will be compared with the characteristics of vehicles detected at the next LPU. If the vehicle features are the same, then the car is successfully identified and successfully tracked. Vehicle features can also be used to track drivers who commit traffic violations. Vehicle features can also be used to monitor the latest real conditions of highway traffic. To demonstrate how well SVS works, we experimented with tracking vehicle types using Kalman Filter. The reason for choosing Kalman Filter because this method can predict the best location for the next frame [6].

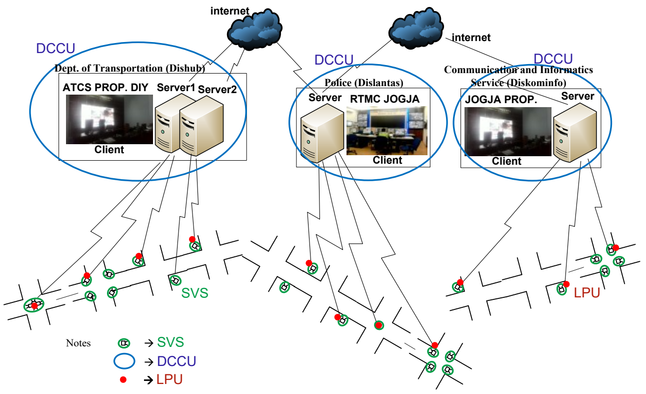

Fig. 3. The proposed Yogyakarta Traffic Monitoring and Management System

II. Vehicle Tracking Model Based on SVS Architecture

Fig. 4 shows the proposed SVS Architecture. The SVS architectural components are Video Sensor, Video Compression Module, Feature Extraction Module, Feature Representation Module, and Data Transmission Module. SVS works to extract the features of traffic objects, compress the video and deliver its processing to DCCU (Data Center and Control Unit) or LPU (Local Processing Unit). SVS input is a scene. The Video Sensor Module capture the scene by streaming frame per frame. Video Compression Module and Feature Extraction Module receive digital video data. The Video Compression Module works to compress video data to minimize the size of a video byte. The feature extraction module works to extract the video data into frames and extract the features of each frame. The Feature Representation Module works to form object features in XML format. The Data Transmission Module serves to transmit compressed video data and object features in XML format to DCCU or LPU. SVS output is a feature of compressed traffic and video data.

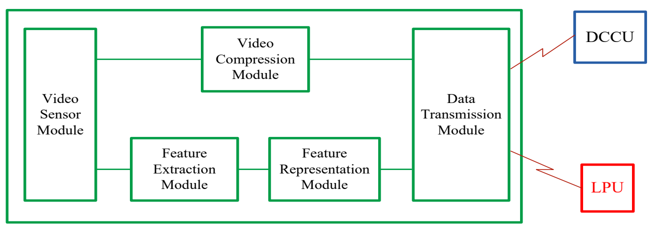

Fig. 4. The proposed SVS Architecture [7]

These features of traffic objects and compressed video data are received at DCCU. These features are synchronized with compressed video data using a time stamp. These features become the basis for further data processing. In DCCU, other systems process the characteristics of traffic objects according to the application needs in that place.

This study is a simulation of how Smart Video Sensor works. The video data format is obtained from DCCU is 3GP. This video comes from 25 CCTV locations in Yogyakarta. The duration of each video is 15 minutes with a frame rate of 30 fps. All data processing is done in a computer that functions as a Smart Video Sensor. The sending feature is delivered to the machine itself that serves as an LPU. This research produces a Smart Video Sensor Architecture that focuses on Feature Extraction Module and Feature Representation Module. Fig.5 shows an SVS workflow focusing on the Feature Extraction Module and Feature Representation Module. The Feature Extraction module performs pre-processing, segmentation, and feature extraction. The Feature Representation Module shows a representation process that transforms features into XML format using a representation scheme [7].

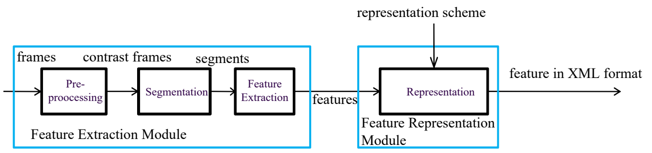

Fig. 5. SVS workflow focusing on the Feature Extraction Module and Feature Representation Module [7]

Initially, the video sensor automatically records the image streaming frame per frame. To improve the quality of each frame then pre-processing uses Histogram Equalization (HE) method. The reason for choosing HE is because this method can improve the overall image quality [8]. Fig. 6 shows before and after using HE.

Fig. 6. Frame before and after HE

After that, each frame is segmented into segments using the Gaussian Mixture Model (GMM) method. The reason for choosing GMM is because it is more adaptive to background changes of objects [9]. Each segment is subsequently labeled. Fig. 7 shows before and after using GMM.

Fig. 7. Frame before and after segmentation using GMM

Segmentation also serves as post-processing. The post-processing goal is to eliminate noise. The post-processing applied is the opening and closing morphology. Opening morphology generally thickens shape lines, which is an erosion operation followed by a dilated operation using the same structural element. Closing morphology attenuates shape lines as well as removing small holes and fills gaps in contour lines, that is carried out by performing a dilated operation followed by erosion operations [10]. Post-processing input data are segments, while the post-processing results are fully loaded parts. Fig. 8 shows before and after using the opening and closing morphology.

Fig. 8. Frame before and after using the opening and closing morphology

The segment extracts 39 features consisting of: Bounding Box, Width of the bounding box (WBB), Height of the bounding box (HBB), Major Axis, Minor Axis, Perimeter, Extent, Eccentricity, Filled Area, Dispersedness, Equivdiameter, Solidity, Convex Area, Circularity [11], color Moment [12], Orientation [13], Centroid [3], Axis Ratio, Gray HOG [14], Area ROI, Aspect Ratio, Area Ratio [15], Object Area [16], Invers Dispersedness [3], Edge Density [17], PHOG [18], Euler Number [19], Elongation [20], Haar like feature [21], Gabor [22], Scale Invarient Feature Transform (SIFT) [23], Convex Hull, Invariant Moment [10], Local Binary Pattern (LBP) [24], Statistical Texture [25], color HOG [26], color Histogram [27], color Correlogram [28], color Energy [29]. The reason for the object being extracted is 39 features because these features will be used for object identification, object detection, object recognition, object classification, object tracking, calculation of the number of objects, calculation of object speed, estimation of traffic jams and other traffic parameters required by the Police, Diskominfo, and Dishub. The features are represented in tabular form. The table describes a sequence of objects scheme that contains features. The schema of the feature representation generates an XML schema [30]. The reason for the use of XML structures is because the adaptive XML structure is exchanged in different platforms [31].

The characteristics of traffic objects and SVS concepts need to be tested. This features of a traffic object in XML format are tested to prove that the elements are correct and efficient. SVS is examined to demonstrate that the SVS concept works well. Therefore, the vehicle model of vehicle tracking is made according to its class type. This tracking model uses the Kalman Filter method. So, features in XML format sent from SVS to DCCU (Data Center and Control Unit). Fig. 9 shows the vehicle tracking model based on the SVS Architecture that works in the DCCU area. This model was created to demonstrate that the SVS concept works well. This model is divided into two processes namely the training phase and the testing phase.

A. Training Phase

The training phase in Fig. 9 performs pre-processing, segmentation and feature extraction. Initial data is in the form of digital video data. Video data is then converted into frames. After that, to improve the qualities of each frame, the pre-processing uses Histogram Equalization (HE) method. Frames are segmented using the Gaussian Mixture Model (GMM). Segmentation is used to separate the object background with an object and to separate the object from one object to another. GMM is used to model each pixel of a sequential image into an updated model distribution based on the difference between the current frame and the previous frame. Segmentation results are segments. The next process is extracting features Local Binary Pattern (LBP), AreaROI and aspect ratio. Feature extraction is used to retrieve unique values of an object that distinguishes it from other purposes [32]. The LBP feature was chosen because it has 59 dimensions so that it gives a better classification and can differentiate uniquely between one object and another. The Area ROI feature is selected because the area can be used to differentiate vehicle classes. The aspect ratio feature is chosen because the difference in value ratio can distinguish the vehicle class. The next step is the classification. The classification method used is K-Nearest Neighbor (KNN) because of its excellent accuracy [33]. KNN is a method of classifying objects based on training data closest to the object [33]. The class of vehicles is motorcycles, cars, buses, and trucks in line with government regulation no. 55 of 2012 on vehicles [34] and levels of accuracy as shown by some research [33]. To find the best K value is done KNN train. The purpose of determining the best value of K is to obtain the best accuracy results. The best value obtained shows the closest neighbor distance uses Euclidean distance. The best K value is 7.

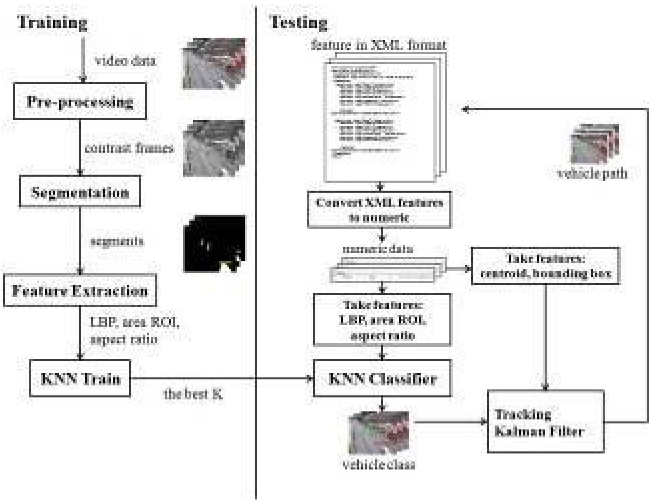

Fig. 9. Vehicle Tracking Block Diagram utilizes the object features of SVS

B. *Testing Phase*

The testing phase from Fig. 9 shows that the test data is the features in XML format. This test data comes from SVS. These features convert its format from XML to numeric to form numeric data tables. Then, taken features LBP, Area ROI and Aspect Ratio to get the vehicle class uses KNN Classifier. After that, search the closest neighbor distance from features LBP, Area ROI and Aspect Ratio to obtain vehicle class.

The next process is the state matrix leads to predicted tracking. The features used for monitoring are centroid and bounding box features. The tracking input for the initial state is the value of centroid and bounding box. Then, the initial state matrix is converted to a previous state matrix subsequently. To ensure the reliability of the next state value then the state needs to be updated using measurement and Kalman Gain. This measurement is derived from the current centroid state matrix. The result is the latest state matrix. After that, the state of the current matrix is converted to the previous state matrix. This result is used to form the vehicle path. The vehicle path is formed from the state of the current matrix reduced by the earlier state matrix. The formation of the vehicle path then completes one tracks cycle. This cycle continues throughout the tracking process.

III. Result and Discussion

Testing of vehicle tracking according to its type based on SVS Architecture is done to demonstrate whether the SVS concept works well. The tests were conducted at three different data locations in Yogyakarta. The site is Galeria, Babar Sari, and Condong Catur. For testing, each location uses frames of 1000 frames. These video frame rates are 30 fps.

Fig. 10 shows the results of vehicle tracking test based on SVS Architecture in Yogyakarta. This recorded image in Fig. 10a shows two motors and one car tracked in Galeria. In Fig. 10b shows two engines traced in Babarsari. In Fig. 10c shows one motor, one car, and one bus followed in Condong Catur.

Fig. 10. Results of vehicle tracking test based on SVS Architecture in Yogyakarta (a) Galeria, (b) Babar Sari, (c) Condong Catur

The number of vehicles marked and counted for classification if the car was once tracked correctly. The accuracy of the vehicle counts from the number of cars followed successfully [35] as shown in (1).

$$Accuracy = \frac{number\ of\ correct\ data}{number\ of\ total\ data} \tag{1}$$

Fig. 11, Fig. 12 and Fig. 13 show experiments with some variations of the number of KNN. The names of videos selected for KNN train are Babarsari_8, Galeria_8, Condongcatur_11. The reason for choosing this video is because: (1) The vehicles are only vehicles exposing the front and rear at several locations intersection of Yogyakarta, (2) The number of variations of vehicle type is the same as the number of vehicle classes used in this study except for Galeria_8 because buses and trucks do not pass in Galeria.

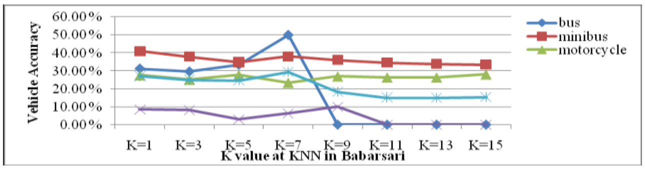

Fig. 11. Relationship between vehicle accuracy and K value at KNN in Babarsari

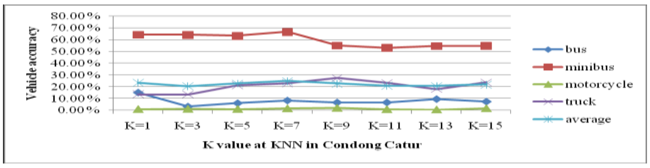

Fig. 12. Relationship between vehicle accuracy and K value at KNN in Condong Catur

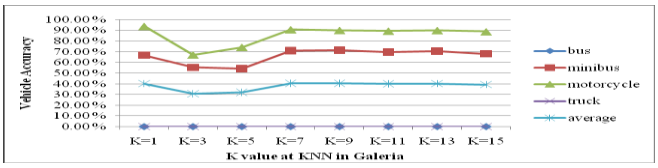

Fig. 13. Relationship between vehicle accuracy and K value at KNN in Galeria

The number of video frames from Babarsari_8, Galeria_8, Condongcatur_11 is 1814 frames, 1000 frames, and 1000 frames, respectively. The best KNN train value is seen from the highest average value of value K. The highest average value of video Babarsari_8, Galeria_8, Condongcatur_11 is 29.42, 80.81, and 24.95 with K = 7, respectively. From some experiments, it can be concluded that the best value of K obtained at K = 7. KNN train uses Euclidean distance for measuring instrument. The reason for choosing Euclidean distance is because of its simplicity [36]. To determine the vehicle class by using the best value of K = 7 is find the majority of 7 nearest neighbors. If the majority of that nearest class is a motor, then the object belongs to the motor class.

Table I shows the results of vehicle class testing based on SVS Architecture. The experimental results show the highest accuracy found on motorcycles are in Galeria (90.71%), for cars in Galeria (70.91%), for buses in Condong Catur (6.25%), and for trucks in Condong Catur (2.44%). The success of this test will assist in further applications. Validation of vehicle tracking based on Kalman Filter needs to be done to ascertain the difference of centroid value derived from the feature in XML format with centroid value generated by Kalman Filter. We use Euclidean distance to calculate the difference by finding the average length of the centroid. The reason for choosing the Euclidean distance is because it is simple in finding the distance between two points [36]. The data samples used for vehicle tracking validation based on the Kalman Filter following the SVS Architecture are from the Galeria location, with the name Galeria_8 video.

Table II and Table III show samples one and two on the difference of centroid point from Kalman Filter and feature in XML format. Both tables show the average difference between the centroid point of the Kalman Filter and that feature in XML format of 0.070 and 0.038 respectively. The result obtained an average value of 0.055. This value indicates that there is a difference of Kalman Filter centroid value of 0.055 compared to the centroid value derived from the feature in XML format.

IV. Conclusion

Testing of vehicle tracking by type based on the SVS Architecture is done to demonstrate that this concept has worked well. Testing has been done. The tests were conducted in 3 different locations in Yogyakarta: Galeria, Babar Sari, and Condong Catur. In this paper, the video data used for each area includes 1000 frames. The highest accuracy found for motorcycles is in Gondomanan (90.71%), for cars in Galeria (70.91%), for buses in Condong Catur (6.25%) and trucks in Condong Catur (2.44%). The tracking test presented in this paper uses two samples of the centroid. The result obtained an average value of 0.055. This value indicates that there is a difference of Kalman Filter centroid value of 0.055 compared to the centroid value derived from the feature in XML format. Thus, other features can be used for various applications as needed. In the subsequent research, SVS testing can be used to determine the number of vehicles, calculate vehicle speed and calculate the density of cars at a particular location. In the next study, it is possible to select features to obtain the best features for classification. In further research, vehicle classification can use Support Vector Machine (SVM), Bayesian Network, Naive-Bayes or Random Forest to improve accuracy results.

TABLE I. THE ACCURACY OF VEHICLE CLASS BASED ON SVS ARCHITECTURE IN%

Location	bus	minibus	motorcycle	truck
Galeria_8	-	70.91	90.71	-
Babarsari_8	-	17.51	37.33	-
Condong catur_11	6.25	65.73	7.76	2.44

TABLE II. SAMPLE 1 DIFFERENCE OF CENTROID POINT FROM KALMAN FILTER AND FEATURE IN XML FORMAT

No	Frame no	Object no	centroid	centroidKF	distance
1	53	1	(233, 167)	(233, 167)	0
2	54	1	(234, 167)	(233.97, 167)	0.0278
3	55	1	(234, 167)	(234.00, 167)	0.0058
4	56	2	(236, 171)	(235.90, 170.79)	0.22754694
5	57	2	(236, 172)	(236.04, 172.01)	0.03848324
6	58	2	(236, 173)	(236.03, 173.02)	0.03597235
7	59	2	(237,173)	(236.97, 173.07)	0.07303164
8	60	1	(237,173)	(237.03, 173.05)	0.05497281
9	61	2	(238,175)	(237.97, 174.92)	0.08298825
10	62	2	(239,173)	(238.98, 173.15)	0.15412579
11	63	2	(240,173)	(239.99, 173.01)	0.01528561
12	64	2	(241,174)	(240.99, 173.94)	0.05677059
13	65	1	(241,177)	(241.05, 176.86)	0.15043234
14	66	2	(242,177)	(241.98, 177.06)	0.06290191
				Average=	0.07043653

TABLE III. SAMPLE 2 DIFFERENCE OF CENTROID POINT FROM KALMAN FILTER AND FEATURE IN XML FORMAT

No	Frame no	Object no	centroid	centroidKF	distance
1	56	1	(181, 187)	(181, 187)	0
2	57	1	(182, 187)	(181.97, 187)	0.0278
3	58	1	(182, 188)	(182.01, 187.95)	0.0528194
4	59	1	(183, 189)	(182.95, 188.96)	0.0581237
5	61	1	(184, 190)	(183.99,190)	0.0052345
6	62	1	(183, 191)	(183.08, 190.98)	0.0802373
7	63	1	(184, 192)	(183.95, 191.98)	0.0528486
8	64	1	(184, 192)	(184.01, 192.04)	0.044091
9	66	1	(185, 194)	(184.99, 193.98)	0.0201405
				Average=	0.0379217

References

[1] Kemenhub, "Peraturan Pemerintah Republik Indonesia Nomor 32 Tahun 2011 tentang Manajemen dan Rekayasa, Analisis Dampak, serta Manajemen Kebutuhan Lalu Lintas," 2011.

[2] M. N. Munawar, R. Sarno, and D. A. Asfani, "SIGNIFICANT PREPROCESSING METHOD IN EEG-BASED," vol. 87, no. 2, pp. 176–190, 2016.

[3] J. Sánchez, G. Benet, and J. E. Simó, "Video Sensor Architecture for Surveillance Applications," *Sensors (Basel).*, vol. 12, no. 2, pp. 1509–1528, Jan. 2012.

[4] T. V. Mathew and K. V. K. Rao, *Introduction to Transportation Engineering: Chapter 30. Fundamental parameters of traffic flow*. India: NPTEL, 2006.

[5] T. R. Board, *Highway Capacity Manual*. United States of America: National Research Council, 2000.

[6] Y. Wang, X. Wu, X. Li, and J. Zhou, “Vehicle type recognition in sensor networks using improved time encoded signal processing algorithm,” *Math. Probl. Eng.*, vol. 2014, pp. 1–8, 2014.

[7] I. Imelda, A. Harjoko, and P. Nurwantoro, “Pemodelan Ekstraksi Ciri Objek Lalu Lintas pada Smart Video Sensor (Feature Extraction Modeling for Traffic Objects on Smart Video Sensor),” Universitas Gadjah Mada, 2018.

[8] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real-time video system," *Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014*, pp. 2392–2397, 2014.

[9] B. Basri, I. Bayu, and A. Achmad, “Gaussian Mixture Models Optimization for Counting The Numbers of Vehicle by Adjusting The Region of Interest under Heavy Traffic Condition,” in *2015 International Seminar on Intelligent Technology and Its Applications*, 2015, pp. 245–249.

[10] R. C. Gonzalez, R. E. Woods, and B. R. Masters, *Digital Image Processing, Third Edition*, vol. 14, no. 2. Prentice-Hall, Inc., 2008.

[11] Z. Chen, T. Ellis, and S. A. Velastin, “Vehicle Type Categorization: A Comparison of Classification Schemes,” in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 74–79.

[12] V. Vinayak and S. Jindal, “CBIR System using Color Moment and Color Auto-Correlogram with Block Truncation Coding,” *Int. J. Comput. Appl.*, vol. 161, no. 9, pp. 1–7, 2017.

[13] C. Chun-Hong and S. Zhi-Lei, “Improved LBP Vehicles Texture Features Extraction Algorithm in Video Monitor Systems,” *2016 Int. Conf. Virtual Real. Vis.*, pp. 87–93, 2016.

[14] N. C. Mithun, N. U. Rashid, and S. M. M. Rahman, “Detection and Classification of Vehicles From Video Using Multiple Time-Spatial Images,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1215–1225, Sep. 2012.

[15] D. Huang, C. Chen, W. Hu, S. Yi, and Y. Lin, “Feature-Based Vehicle Flow Analysis and Measurement for a Real-Time Traffic Surveillance System,” *J. Inf. Hiding Multimedia. Signal Process. Ubiquitous Int.*, vol. 3, no. 3, pp. 282–296, 2012.

[16] L. Yang, J. Johnstone, and C. Zhang, “A Multi-camera Approach to Vehicle Tracking Based on Features,” *Ninth IEEE Int. Symp. Multimedia. Work. (ISMW 2007)*, pp. 79–80, Dec. 2007.

[17] C. Bravo, N. Sánchez, N. García, and J. M. Menéndez, “Outdoor vacant parking space detector for improving mobility in smart cities,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8154 LNAI, pp. 30–41, 2013.

[18] S. K. Patel and A. Mishra, “Moving Object Tracking Techniques : A Critical Review,” *Indian J. Comput. Sci. Eng.*, vol. 4, no. 2, pp. 95–102, 2013.

[19] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*, Second Edi. 2009.

[20] B. T. Morris and M. M. Trivedi, “Learning, modeling, and classification of vehicle track patterns from live video,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 425–437, 2008.

[21] S. P. Adhikari, H. J. Yoo, and H. Kim, "Real-time on-road vehicle detection with low-level visual features and boosted cascade of Haar-like features," *J. Inst. Control. Robot. Syst.*, vol. 17, no. 1, pp. 17–21, 2011.

[22] A. Nurhadiyatna, A. L. Latifah, and D. Fryantoni, “Gabor filtering for feature extraction in real time vehicle classification system,” *9th Int. Symp. Image Signal Process. Anal. ISPA 2015*, no. Ispa, pp. 19–24, 2015.

[23] P. K. Mishra and B. Banerjee, "Multiple Kernel-Based KNN Classifiers for Vehicle Classification," *Int. J. Comput. Appl.*, vol. 71, no. 6, pp. 1–7, 2013.

[24] D. Neumann, T. Langner, F. Ulbrich, D. Spitta, and D. Goehring, "Online vehicle detection using Haar-like, LBP and HOG feature-based image classifiers with stereo vision preselection," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 773–778, 2017.

[25] O. B. Abouelatta, “Classification of Copper Alloys Microstructure using Image Processing and Neural Network,” *J. Am. Sci.*, vol. 9, no. 6, pp. 213–223, 2013.

[26] S. Srivastava and P. Singh, “Real-Time Object Tracking Using Colour Feature,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 3, no. 8, pp. 56–59, 2014.

[27] R. Kavya and Harisha, “Feature Extraction Technique for Robust and Fast Visual Tracking : A Typical Review,” *Int. J. Emerg. Eng. Res. Technol.*, vol. 3, no. 1, pp. 98–104, 2015.

[28] Y.-C. Wang, C.-C. Han, C.-T. Hsieh, and K.-C. Fan, “Vehicle color classification using manifold learning methods from urban surveillance videos,” *EURASIP J. Image Video Process.*, vol. 2014, no. 1, p. 48, 2014.

[29] F. Jialiang, D. Jiabi, and Z. Sifa, “Combination of Wavelet Transform and Mutual Information for Vehicle Recognition,” in *International Congress on Sound & Vibration*, 2016, no. July, pp. 1–8.

[30] I. Imelda, A. Harjoko, and P. Nurwantoro, “Feature Representation Scheme for Smart Video Sensor,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 1, pp. 1–9, 2018.

[31] J. J. Parsons and D. Oja, *New Perspective on Computer Concepts 2012*. United States of America: Cengage Learning, Inc., 2012.

[32] Y. Mingqiang, K. Kidiyo, and R. Joseph, “A survey of shape feature extraction techniques,” *Pattern Recognit.*, vol. 2008, pp. 43–90, 2008.

[33] F. M. D. S. Matos and R. M. C. R. De Souza, “An image vehicle classification method based on edge and PCA applied to blocks,” *2012 IEEE Int. Conf. Syst. Man, Cybern.*, pp. 1688–1693, Oct. 2012.

[34] Kemenhub, “Peraturan Pemerintah Republik Indonesia Nomor 55 Tahun 2012 tentang Kendaraan,” 2012.

[35] A. Salarpour, A. Salarpour, M. Fathi, and M. Dezfoulian, “Vehicle Tracking using Kalman Filter and Features,” *Signal Image Process. An Int. J.*, vol. 2, no. 2, pp. 1–8, 2011.

[36] C. Shi, W. S. Gan, Y. K. Chong, A. Apoorv, and K. S. Song, “A vehicular noise surveillance system integrated with vehicle type classification,” *2013 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA 2013*, 2013.