A Robust Multiclass Vehicle Detection and Classification Algorithm for Traffic Surveillance System

Long Hoang Pham*
Sungkyunkwan University
Suwon, South Korea
phlong@skku.edu

Hung Ngoc Phan*
International University,
Vietnam National University
Ho Chi Minh City, Vietnam
hungpn17@mp.hcmiu.edu.vn

Nhat Minh Chung
International University,
Vietnam National University
Ho Chi Minh City, Vietnam
ITITIU17012@student.hcmiu.edu.vn

Tuan-Anh Vu
The Hong Kong University of
Science and Technology
Kowloon, Hong Kong
tuananh.vu@connect.ust.hk

Synh Viet-Uyen Ha†
International University,
Vietnam National University
Ho Chi Minh City, Vietnam
hvusynh@hcmiu.edu.vn

***Abstract*—The main goal of traffic surveillance systems (TSSs) is to extract useful traffic information by analyzing signals from cameras. This paper presents a system for vehicle detection and classification from static pole-mounted roadside surveillance cameras on busy streets in the presence of different kinds of vehicles. There has been considerable research to accommodate this subject since the 90s; but most studies have been only carried out in developed countries where traffic infrastructures are built around automobiles, whereas in developing countries, motorbikes are dominant. This paper proposes a method that robustly detects, classifies and counts vehicles into three classes: light (motorbikes, bikes, tricycles), medium (cars, sedans, SUV), heavy vehicle (trucks, buses), and a novel tracking algorithm designed to enable classification by majority voting to cope with motorbikes' sudden changes in direction. Extensive experiments with real-world data to evaluate the system's performance have shown promising results: a detection rate of 95.3% in daytime scenes.**

***Index Terms*—Vehicle detection, vehicle classification, vehicle tracking, real-time traffic surveillance system.**

I. Introduction

In recent years, there have been increasing interests in the area of traffic surveillance system (TSS), especially in Vietnam and other developing countries. The main goal of a TSS is to gain an understanding of traffic situations through an extraction of information (counts, speed, vehicle type, and density) from sensors' signals. So far, many studies have been carried out using cutting-edge TSSs. However, the detectors in use are costly, bulky and are difficult to maintain, while still providing limited information [1]. Perhaps for those reasons, video-based TSSs are becoming more popular. They are capable of providing more information about traffic conditions and can adapt to a wide range of view condition. They are characterized as low cost, less disruptive and more maintainable than others.

Last decade has seen a massive amount of research into image analysis for vehicle classification. In early work, Wang et al. [2] performed studies on vehicle classification at intersections with a 2-step algorithm. They first validate vehicle using the object weight, height, aspect ratio, area criteria, extracting the texture features using the HOG descriptor; then a multi-class SVM classifier is trained to identify vehicle types. Chen and Ellis [3] combined a set of measurement features with a pyramid of HOG features (both edge and intensity) to classify vehicles into four categories: car, van, bus, and motorbike. However, in their experiments, motorbikes only contribute small sample sizes. Regarding supervised approaches with neural networks, Dong et al. [4] proposed a vehicle classification method using a semi-supervised convolution neural network on vehicle frontal view images. Hao et al. [5] used extreme learning machine (ELM) on invariant moments and horizontal edge features to classify vehicles into sedan, bus, and van truck. Despite the amount of literature, vehicle classification has not yet received much attention concerning 2-wheeled vehicles. This may be because of the lack of scenarios where previous studies took place.

In this paper, we present a multi-class vehicle detection and classification algorithm that works robustly in daytime surveillance scenes, and is explicitly designed to cope with the complex behavior of 2-wheeled motorized vehicles. In summary, the main contributions of this work are: 1) an approach utilizing decision tree analysis on measurement-based features (dimension ratio, density ratio, and size) to achieve high levels of vehicle classification performance; and 2) a new tracking method to demonstrate the effectiveness of vehicle labeling in chaos traffic scenes.

The paper is organized as follows: Section II-A presents a method for vehicle detection. Section II-B and Section II-C respectively describe our proposed method of feature extraction and vehicle classification. Our new approach for vehicle tracking is then characterizes in Section II-D. The rest of the paper includes experiment results and discussions.

II. Proposed Method

A. Vehicle Detection

The effectiveness of the background model is evaluated through a binary foreground mask in which moving objects are marked as white blobs. However, in urban traffic environment, a real background is not always available and can be influenced

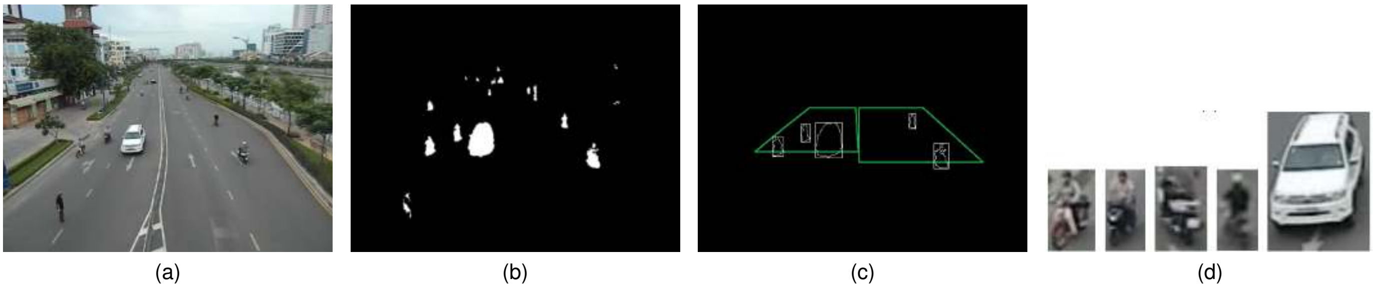


Fig. 1. Illustration of vehicle detection process. (a) Original image. (b) Foreground image from background subtraction process. (c) Extract vehicles' contours inside observation zone. (d) Extracted vehicle images.

by objects being introduced or removed from the scene, slow-moving or stationary objects, camera vibration (e.g. strong wind, heavy vehicles). In order to account for these problems, we adopt the background subtraction algorithm proposed by Nguyen et al. [6] and Ha et al. [7] to extract moving objects. The novelty of this technique is the application of an entropy function (EF) in a Gaussian mixture model (GMM) and disorder removal framework to neglect the disorder frames (DF), thus, selecting proper images to model the background and to detect foreground components.

With the constructed background, moving foreground objects are extracted with pre-processing and filtering operations, which aim to remove shadow [8], on-road reflections in rainy scenes [9], and noises are executed to refine the vehicles' images. Then objects' contours residing inside observation zones are extracted using [10]) as shown in Fig. 1.

B. Feature Extraction

From extricated vehicles, we obtain the set of vehicle candidates, with a list of measurement features that represent the ith vehicle at the kth frame, as described in Table I.

TABLE I
VEHICLE MEASUREMENT FEATURES

Symbol	Description
$B_i^h(k)$	Height of vehicle's bounding box
$B_i^w(k)$	Width of vehicle's bounding box
$E_i^h(k)$	Major axis of vehicle' bounding ellipse
$E_i^w(k)$	Minor axis of vehicle's bounding ellipse
$P_i^E(k)$	Total pixels inside vehicle's bounding ellipse
$P_i^C(k)$	Total pixels inside vehicle's contour (vehicle area)
$P_i^{CH}(k)$	Total pixels inside vehicle's Convex Hull contour
$L_i^C(k)$	Perimeter of vehicle's contour
$R_i^{di}(k)$	Dimension ratio of vehicle
$R_i^{de}(k)$	Density ratio of vehicle

Working with simple features makes a solution more competitive in terms of computation and storage. Thus, we select an optimal feature subset with principle component analysis (PCA) as we discard variables that gives little extra information. Namely, we focus on reducing the original variables into a lower number of non-correlated synthesized variables, then mathematically determining the linear combinations of variables which preserve the information of data. Figure 2 illustrates the variance (eigenvalue) percentages suggesting how much information each of the 10 principle components (PCs) can describe. We can see that the first three PCs accounts for 98.5% of the total variance, while the rest accumulate for just a small portion.

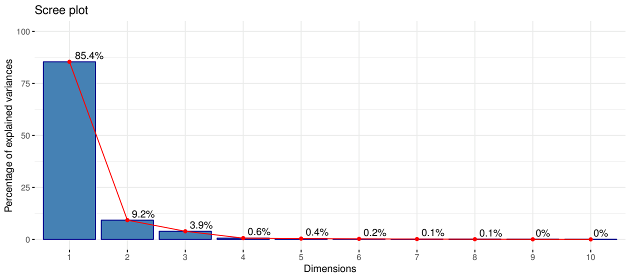


Fig. 2. Scree plot of principle components using variances.

Table II summarizes the loading that each variable contributes to the first three PCs. We then observe that contour area contributes the most to PC1, dimension ratio contributes the most to PC2, and density ratio contributes the most to PC3. So the feature subset that is enough to account for 98.5% of the total variance can be:

- 1 – dimension ratio, $R_i^{di}(k)$.
- 2 – density ratio, $R_i^{de}(k)$.
- 3 – contour area/vehicle size, $P_i^C(k)$.

Blob analysis then identifies non-vehicle objects and removes them. As the camera is mounted on an elevated platform looking straight down on the road, the rotation angle E° of a vehicle' bounding ellipse with respect to the horizontal plane is approximately 90°. So if the ith object satisfies one condition, it is considered a vehicle candidate: $85^\circ \leq E_i^\circ(k) \leq 105^\circ$, where the lower and upper values are empirically selected.

TABLE II
SUMMARY OF VARIABLES' LOADINGS TO THE FIRST THREE PCS.

Variable	PC1	PC2	PC3
$P_i^E(k)$	0.3345422	-0.1562512	0.05551648
$E_i^h(k)$	0.3278795	-0.2204513	0.17096738
$E_i^w(k)$	0.3351614	0.148366	-0.16941252
$\boldsymbol{R_i^{di}(k)}$	0.2433402	**0.6454741**	-0.51207482
$\boldsymbol{R_i^{de}(k)}$	0.2356265	0.5603481	**0.77562315**
$P_i^{CH}(k)$	0.3361558	-0.1767915	0.01284477
$L_i^C(k)$	0.3283365	-0.1848338	-0.11267756
$\boldsymbol{P_i^C(k)}$	**0.3375633**	-0.126193	0.05073695
$B_i^h(k)$	0.3271465	-0.2803197	0.05786459
$B_i^w(k)$	0.3342598	0.1201035	-0.23746961

C. Vehicle Classification

The vehicle classification algorithm is the main focus of this paper, which is a extension of our prior work [11], [12]. In our algorithm, vehicles are classified into three classes:

- **Class 1**: Light vehicles (motorbikes and bicycles).
- **Class 2**: Medium vehicles (cars, sedans and SUVs).
- **Class 3**: Heavy vehicle (trucks and buses).

Three observations can be made: 1) light vehicles have high eccentricity bounding ellipses that resembles human shapes; 2) the bounding ellipses of heavy vehicles share these traits but with larger sizes; 3) medium vehicles' shapes are squarer than others and they cover extra pixels belonging to the background. Vehicle size can be obtained by counting the total pixels of the vehicle. The eccentricity of an ellipse can be measured by its ratio between the minor and major axis, or namely, by a dimension ratio, R^{di}. In order to implement the third observation, we propose a density ratio, R^{de}, which is the ratio of total pixels of the vehicle and the total pixels of the bounding ellipse:

$$R_i^{di}(k) = \frac{E_i^w(k)}{E_i^h(k))} \tag{1}$$

$$R_i^{de}(k) = \frac{P_i^C(k)}{P_i^E(k)} \tag{2}$$

Several experiments have been performed, in each of which we obtain the distribution of three vehicle classes on a 3-axis scatter plot just like the one shown in Fig. 3. The results further confirm our observations as we can conclude that light vehicles and medium vehicles are separated along the R^{di}–R^{de} plane, not to mention that medium vehicles are usually double in size. Meanwhile, heavy vehicles are scattered across the R^{di}–R^{de} plane. A possible explanation is that, for instance, 16-seat buses and delivery trucks have similar shapes to medium vehicles whose bounding ellipses have high eccentricity values; while 50-seat buses and trailers, because of their length, have thinner bounding ellipses, and thus share traits with light vehicles [13]. However, their sizes distinguish them from other classes.

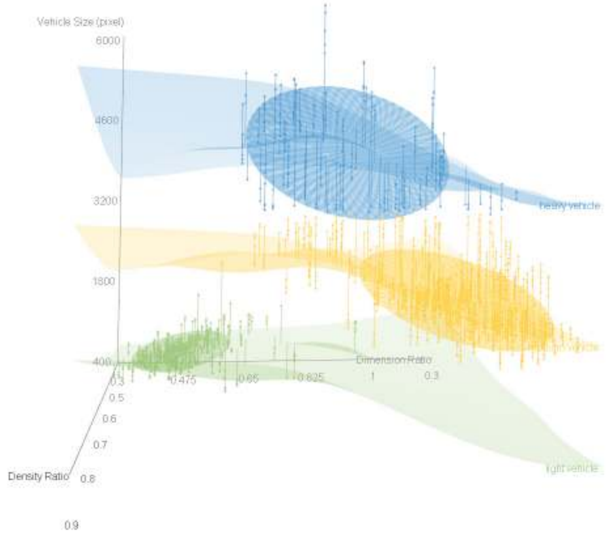


Fig. 3. The distribution of vehicles for dataset PVD1. Light, medium and heavy vehicles are shown as green, yellow and blue dots respectively.

Once vehicles' features are obtained, we use the decision tree theory proposed by [14] to produce the labeling rules. Decision tree builds classification models in the form of a tree structure which can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one. The core algorithm for building decision trees called ID3 employs entropy and information gain to construct the tree in a top-down approach. From a root node, data is partitioned into smaller and smaller subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is equally divided it has entropy of one. The information gain is calculated based on the decrease in entropy after a dataset is split on an attribute. The attribute maximizing the information gain the most is selected as the root node. The value of the attribute at which this gain occurs is obviously the split point. This partitioning process may continue along the same attribute to yield smaller and smaller subsets of higher purity. Fig. 4 shows the decision tree of dataset PVD1.

In our intensive analysis, K-fold cross validation is applied to check the integrity of the decision model. The k-fold cross validation method involves splitting the dataset into k-subsets. For each subset, it is held out while the model is trained on all other subsets. This process is completed until accuracy is determined in each instance of the dataset, and an overall accuracy estimate is provided. We use 10-fold cross validation to estimate the accuracy of the decision model. The overall accuracy is 96.86% with a standard deviation of 1.41%. In addition, the Kappa statistic for the model is computed. The kappa value is a metric that compares an observed accuracy with an expected accuracy (random chance). In other words, the kappa statistic is a measure of how closely the instances classified by the machine learning classifier matched the data labeled as ground truth. The overall kappa value in our test is

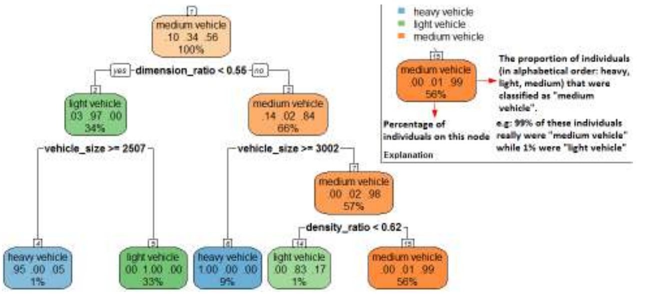


Fig. 4. The decision tree for vehicle labeling of dataset PVD1. The label at the top each node signifies the vehicle type. The three numbers in the middle of each node are the proportion of individuals with respect to heavy, light, and medium vehicle that were actually classified as the label. The number at the bottom of each node is the percentage of individuals.

94.60% with the standard deviation of 2.38%. It shows that our labeling model is somewhat accurate.

D. Vehicle Tracking

Since the aforementioned vehicle detection is performed on consecutive frames, multiple instances of the same vehicle are extracted and classified individually. Therefore, a tracking procedure is applied to analyze the motion of potential vehicles, based on successive image frames. Because vehicles mostly move vertically, their horizontal components change slightly, and their vertical components are in accordance with their speeds. So, if two vehicles (i and j) in two consecutive frames ($C_i(k)$, $C_j(k-1)$) satisfy the following conditions, they are considered as the same vehicle:

1) The horizontal distance between their centroids is smaller than a certain threshold.

$$\left|C_i^x(k) - C_j^x(k-1)\right| \leq TH_1 \tag{3}$$

2) The vertical distance between their centroids is smaller than a certain threshold.

$$\left|C_i^y(k) - C_j^y(k-1)\right| \leq TH_2 \tag{4}$$

3) The vehicle i in the current frame has approximately the same size as the vehicle j in the previous frame.

$$\frac{\min(P_i^C(k), P_j^C(k-1))}{\max(P_i^C(k), P_j^C(k-1))} \geq TH_3 \tag{5}$$

where $C_i^x(k)$ is horizontal coordinate of vehicle's centroid, and $C_i^y(k)$ is vertical coordinate of vehicle's centroid; TH_1 is the maximum horizontal displacement of vehicles in two consecutive frame, TH_2 is the maximum vertical displacement of vehicles in two consecutive frame, and TH_3 is the similarity between vehicle i and j.

There are four possible scenarios after matching $C_i(k)$ and $C_j(k-1)$:

- If $C_i(k)$ is not matched, this is a newly appearing vehicle so $C_i(k)$ is marked as a new vehicle. Note that any new vehicle is given the default "unidentified" label.
- If $C_i(k)$ is matched with $C_j(k-1)$, this is a previously tracked vehicle and therefore the label of $C_j(k-1)$ is assigned to $C_i(k)$. In addition, if the trajectory path of $C_i(k)$ has a length of 2, the vehicle will be relabeled as "vehicle candidate".
- If $C_i(k)$ is a "vehicle candidate", it will be classified and relabeled to one of the three vehicle classes after crossing the counting line (CL). Only vehicle candidate with a minimum of ten trajectory points is considered for counting.
- If $C_j(k-1)$ is not matched, this vehicle has already exited so its label is removed. Note that the vehicle along with its feature vectors will only be deleted if they will have not been matched after $(k+5)$ frames later.

In many cases during the motion of an object, the tracking algorithm suddenly ceases tracking and considers it a new object. Furthermore, some trackers cannot keep track of vehicles' trajectories when they move closely with each other as observed in Fig. 5(a). These are the main reasons for missed counting, as well as duplicate counting. In addressing this problem, we extend the matching process of the tracker to look back up to 5 previous consecutive frames. Note that TH_1 and TH_2 are multiplied by m, where m is the mth look back frame. Fig. 5(b) shows the results of our tracking algorithm.

III. Experiments and Discussion

In order to evaluate the system, experiments were performed on traffic datasets captured in Ho Chi Minh City, Vietnam at several locations from single pole-mounted cameras during daytime periods. The capture rate is 30 frames per second (fps) with the resolution of 640×480. The system has been developed and tested on a computer comprising of Intel Core i7 4710HQ and 8GB of RAM. The experiments were conducted to measure the classification accuracy of the proposed method and to test the system performance as a whole.

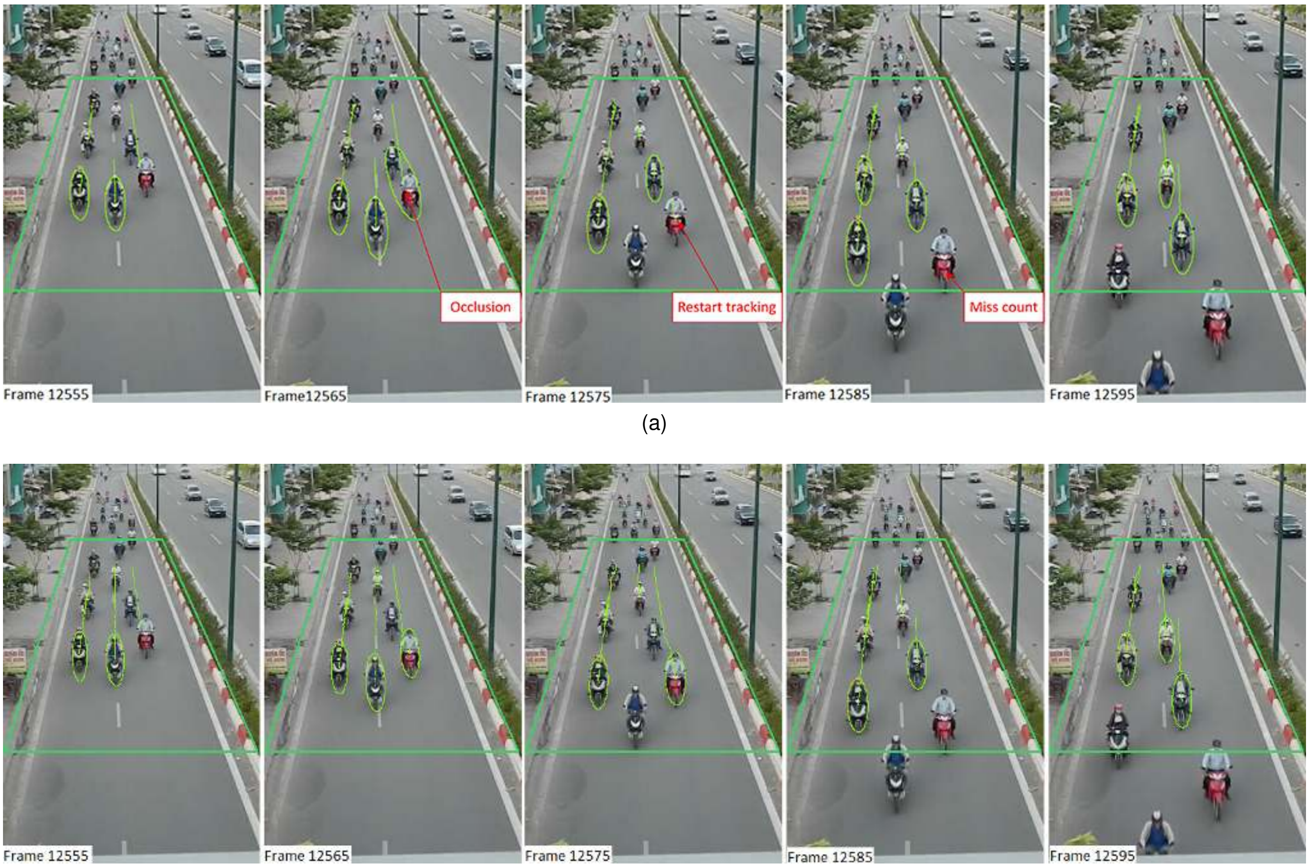
A. Classification Accuracy

Table III summarizes the results of Ha's method [11] and our system on the dataset: VVK1, VVK2, PVD1, and PVD2. These datasets represent ideal traffic environments in developing countries as well as tropical regions. The table consists of the actual number of vehicles, the number of vehicles counted by the system, true positive (TP), true negative (TN), false positive (FP), false negative (FN). The per-class accuracy and recall and total accuracy are calculated as suggested by [15].

Firstly, dataset VVK1 was captured when the sky is clear and there are no shadow appearances where the roads are small and do not have dedicated lanes for different vehicle types. The results can be subjectively evaluated on the 1^{st} column of Fig. 6 where vehicles have been correctly classified to their corresponding classes. Table III also suggests that our method has better accuracy (96.3%) and higher recalls as Ha's method misclassified most buses and trucks as light vehicles (recall = 0%), consequently producing many false-positives.

Secondly, dataset VVK2 was captured when faint shadows are present where the roads are bigger and have dedicated lanes for different vehicle types. The 2^{nd} column of Fig. 6 illustrates the classification results of this dataset. We can observe on the left and right images that the system can maintain stable tracking of motorbikes which travels closely together. The tracking and classification of medium and heavy vehicles also yield good results. When comparing with Ha's results, it can be noticed that their system's counting is higher than ours, but as are their per-class FP values. This is caused by completely missing out on heavy vehicles, resulting in lower accuracy and recall. Meanwhile, the effectiveness of our classifier to identify positive labels is high (average recall = 86.91%), as well as the overall result (96.30%).

Finally, dataset PVD01 was captured as strong shadows appear in almost every frame. This problem can be seen on the 3^{rd} column of Fig. 6 where shadows' intensity is constantly changing with respect to moving clouds. Dataset PVD02 was also captured at the same location, but in rainy weather at dusk. The 4^{th} column of Fig. 6 illustrates the highly complex



(a)

(b)

Fig. 5. Comparison between vehicle tracking algorithm from Ha et al. [11] and our through frame 12555 to 12595 of dataset PVD01. Notice that the vehicle bounded by a green ellipse indicates that it has been successful classified and counted. (a) Tracking results from Ha et al. [11]; the tracker cannot separate the two motorbikes moving closely with each other in frame 12555, hence, results in occlusion in frame 12565 and miss counting the red motorbike in frame 12585. (b) Tracking results using our algorithm; the tracker can maintain solid trajectory tracking for each of the two closely moving vehicles.

TABLE III
COMPARISON OF CLASSIFICATION AND COUNTING RESULTS BETWEEN HA'S METHOD [11] AND OUR PROPOSED METHOD.

			Ha's method [11]								Our method							
Dataset	Class	Actual	Count	TP	TN	FP	FN	Acc	Recall	T.Acc	Count	TP	TN	FP	FN	**Acc**	**Recall**	**T.Acc**
	1	684	566	556	46	10	128	81.35%	81.29%		627	626	57	1	58	**92.05%**	**91.52%**	
VVK01	2	53	50	46	556	4	7	98.21%	86.79%	92.59%	61	49	634	12	4	**97.71%**	**92.45%**	**96.30%**
	3	11	0	0	602	0	11	98.21%	0%		11	8	675	3	3	**99.13%**	**72.73%**	
	1	1008	979	858	91	121	150	77.72%	85.12%		918	911	194	7	97	**91.40%**	**90.38%**	
VVK02	2	116	135	91	858	44	25	93.22%	78.45%	86.94%	112	98	1007	14	13	**97.61%**	**88.29%**	**95.43%**
	3	117	0	0	949	0	107	89.87%	0%		106	96	1009	10	21	**97.27%**	**82.05%**	
	1	312	327	206	214	121	106	64.91%	66.03%		285	274	350	11	38	**92.72%**	**87.83%**	
PVD01	2	250	260	214	206	46	36	83.67%	86.60%	74.57%	231	222	401	9	28	**94.39%**	**88.80%**	**94.47%**
	3	139	0	0	420	0	139	75.13%	0%		141	128	495	13	11	**96.29%**	**92.09%**	
	1	0	20	0	93	20	0	82.30%	0%		0	0	103	0	0	**100%**	**100%**	
PVD02	2	108	95	93	0	2	15	84.55%	86.11%	85.71%	98	97	6	1	11	**89.56%**	**89.81%**	**94.98%**
	3	10	0	0	93	0	10	90.29%	0%		7	6	97	1	4	**95.37%**	**60.00%**	
Overall Accuracy			**84.95%**								**95.30%**							

Note that: TP = true positive; TN = true negative; FP = false positive; FN = false negative; Acc = accuracy; T.Acc = total accuracy.

surveillance scenarios which are caused by mirror-reflection of vehicles and their headlights. Nevertheless, we can obtain good classification results thank to the help of filter operations [8] [9] as previously stated in Section II-A. The system can detect the actual moving vehicles in both scenarios and gain far more practical classification results (T.Acc = 94.47% and 94.98%) than Ha's method (T.Acc = 74.57% and 85.71%).

By comparing with the method of Ha et al. [11], whose studies are most relevant to ours, we have shown that our proposed method is significantly better in terms of both accuracy and recall when classifying vehicles. This result can only be achieved by the combined effectiveness of vehicle detection, filters, tracking, and decision rules. The overall accuracy of the proposed method is roughly 95.3%.

B. System Performance

Finally, our method can process sample sequences at around 30 fps on non-shadows conditions. The processing rate is

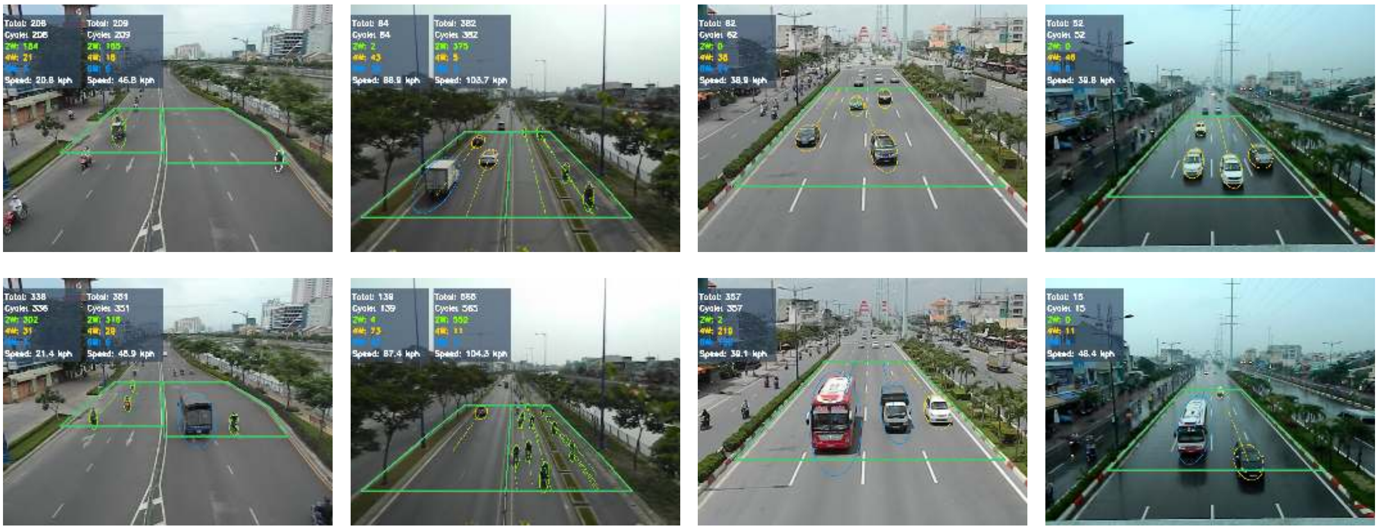


Fig. 6. Examples of vehicle tracking, classification and counting (green ellipses are light vehicles, yellow ellipses are medium vehicles, blue ellipses are heavy vehicles). The 1^{st} column: dataset VVK01 (early morning, mixed lanes). The 2^{nd} column: dataset VVK02 (mid-morning, separate lanes). The 3^{rd} column: dataset PVD01 (afternoon, separate lanes). The 4^{th} column: dataset PVD02 (dusk, separate lanes).

around 28 fps when shadows appear because the system needs to run additional filters. In addition, when running all four videos simultaneously, the system can still retain a frame rate of 25 fps. The tests were performed by calculating the total processing time of 1000 frames. On average, the system is shown to have the ability to process images in real-time. Table IV summarizes the results of these experiments.

TABLE IV
AVERAGE COMPUTING TIME

Dataset	Total frames	Run-time (sec.)	Performance (fps)
VVK01	1000	34.28	29.17
VVK02	1000	30.32	32.98
PVD01	1000	29.52	33.87
PVD02	1000	29.50	33.89
All 4 videos	1000 (x4)	40.67	24.59

IV. CONCLUSION

In this paper, we have presented a vehicle detection and classification system. Our most significant contribution in this study is introducing advancements in vehicle tracking, feature selection, and labeling rules to the classification model. The proposed method presents marked improvements over related studies as it classifies vehicles into three classes: light, medium, and heavy by translating decisions into simple if-else blocks with minimal errors whist keeping track of them. Experiments have confirmed that the system can work robustly throughout the daytime surveillance environments whilst maintaining real-time performance. Based on this framework, future studies can be extended to handle different scenarios such as nighttime, rush-hour, and so on.

REFERENCES

[1] L. Klein, M. Mills, and D. Gibson, *Traffic detector handbook*. U.S. Dept. of Transportation, Federal Highway Administration, Research, Development, and Technology, Turner-Fairbank Highway Research Center, 2006.

[2] Y. Wang, C. Han, C. Hsieh, and K. Fan, "Vehicle type classification from surveillance videos on urban roads," 07 2014, pp. 266–270.

[3] Z. Chen and T. Ellis, "Multi-shape descriptor vehicle classification for urban traffic," in *Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011*, 2011.

[4] Z. Dong, M. Pei, Y. He, T. Liu, Y. Dong, and Y. Jia, "Vehicle type classification using unsupervised convolutional neural network," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 172–177.

[5] Y. shuang Hao, Y.-X. Yin, and J. hui Lan, "Road vehicle classification based on extreme learning machine," 2013.

[6] T. P. Nguyen, C. C. Pham, S. V. Ha, and J. W. Jeon, "Change detection by training a triplet network for motion feature extraction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 433–446, Feb 2019.

[7] S. Viet-Uyen Ha, D. Nguyen-Ngoc Tran, T. P. Nguyen, and S. Vu-Truong Dao, "High variation removal for background subtraction in traffic surveillance systems," *IET Computer Vision*, vol. 12, no. 8, pp. 1163–1170, 2018.

[8] M. Xiao, C. Z. Han, and L. Zhang, "Moving shadow detection and removal for traffic sequences," *International Journal of Automation and Computing*, 2007.

[9] S. V.-U. Ha, N. T. Pham, L. H. Pham, and H. M. Tran, "Robust Reflection Detection and Removal in Rainy Conditions using LAB and HSV Color Spaces," *REV Journal on Electronics and Communications*, 2016.

[10] S. Suzuki and K. A. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics and Image Processing*, 1985.

[11] S. Ha, L. Pham, H. Tran, and P. Ho, "Improved vehicles detection & classification algorithm for traffic surveillance system," *Journal of Information Assurance and Security*, vol. 9, pp. 268–227, 09 2014.

[12] L. H. Pham, T. T. Duong, H. M. Tran, and S. V. Ha, "Vision-based approach for urban vehicle detection amp; classification," in *2013 Third World Congress on Information and Communication Technologies (WICT 2013)*, Dec 2013, pp. 305–310.

[13] R. P. Avery, Y. Wang, and G. S. Rutherford, "Length-based vehicle classification using images from uncalibrated video cameras," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2004.

[14] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[15] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, 2009.