A Real-time Vehicle Detection for Traffic Surveillance System Using a Neural Decision Tree

Hung Ngoc Phan*, Long Hoang Pham*, Tin Trung Thai, Nhat Minh Chung, and Synh Viet-Uyen Ha†

***Abstract*—Traffic surveillance system (TSS) is an essential tool to extract necessary information (count, type, speed, etc.) from cameras for traffic monitoring in many metro cities. In TSS, vehicle detection plays a pivotal role as it is a vital process for further analysis such as vehicle classification and vehicle tracking. So far there has been a considerable amount of research proposed with single-pipeline Convolution Neural Networks (CNN) to accommodate this subject. Although these studies achieved results with high accuracy, they required a large dataset and an implementation on dedicated hardware configuration. This paper presents a novel method with vision-based approach to detect moving vehicles from static surveillance cameras. Moving vehicles are detected and analysed by means of using a Neural Decision Tree accompanied with geometric features to classify vehicles and a Single Shot Detector to handle occlusion when inter-vehicle space between vehicles significantly decreases. Experiments have been conducted on the real-world data to evaluate the performance and accuracy of the proposed method. The results showed that our proposed method achieved a promising detection rate with real-time processing on regular hardware configuration.**

***Index Terms*—Vehicle Detection, Vehicle Classification, Neural Decision Tree, Occlusion Handling, Traffic Surveillance Systems**

I. Introduction

Over the past decade, cutting-edge traffic surveillance systems (TSS) have employed a broad range of advanced sensors for estimating traffic parameters, including magnetic, radar, infrared detectors, inductive loop detectors, and so on. However, such detectors are costly and large in size, have high installation cost and are difficult to maintain, while providing limited information [1]. Accordingly, recent years have witnessed a rapid progression of computer vision, especially expert vision-based systems. Traffic video monitoring systems, which are specific instances in this series, aim to detect, classify vehicles and extract further traffic information. In this context, vehicle detection has emerged as a challenging task because of the importance of localizing vehicles in sequences of static images. Obviously, video-based TSSs gained more superior advantages due to the capability to provide more information about traffic conditions and can be adapted to a wide range of views associated with varying weather [2], illumination conditions and traffic density using sophisticated image analysis tools. Besides, they are low cost, less disruptive, require low maintenance, non-contact.

With the remarkable development of pattern recognition and image processing, a variety of object detection techniques have been proposed. Overall, there are two rudimentary approaches to vehicle detection and classification: model-based and feature-based [3]. In model-based tactics, vehicles are classified based on their shapes, silhouettes and dimensions. A spectacular study conducted by Du *et al.* [4] who transformed each 2D bounding box into a 3D point cloud to detect 3D boundary boxes of vehicles. This method achieves a high vehicle detection rate; however it requires three models to cover all occlusion patterns in implementation. Chabot *et al.* [5] proposed a Deep MANTA that refines 2D bounding boxes of vehicles to recover orientation and 3D location of vehicles via a robust 2D/3D point matching. This manner works well when parts of vehicles are obscured but a dataset of 3D shape templates are required to encode the variability of vehicles. On the other hand, feature-based methods categorize vehicles based on their visual features (including edges, corners, gradients). Ma *et al.* [6] proposed a method of segmenting occluded vehicles by exploiting Difference of Gaussian and Local Binary Pattern features to examine the constraints of symmetry axes of vehicles. Then, a contour concavity analysis is performed to separate vehicles from occlusion blobs. Chang *et al.* [7] present a recursive algorithm combining with high-level visual features extraction to segment vehicles involved in multiple-vehicle occlusions. Velastin *et al.* [8] identified and classified motorcycles in urban environments by employing a Faster R-CNN-like model that was trained and evaluated on 7500 annotated images. The solutions following this approach require a great amount of computation for feature extraction in training phase and are hard to be put into practice without the support of fancy hardware configuration such as graphics processing units (GPU).

In this paper, we propose a novel vehicle detection algorithm that is sufficient to handle occlusion among vehicles with real-time performance. Our work consists of three modules: moving vehicle detection, vehicle classification and occlusion handling. Like motion-based methods, our method exploits a background subtraction model to separate moving vehicles from a static scene. Then, extracted vehicles are examined for classification on a neural decision tree. A model, our main contribution in this work, was developed with a data-driven approach on real-world data based on a soft decision tree presented by Frosst and Hinton [9]. Thereupon, vehicles are classified into three specific classes and occlusion blobs are

S. V.-U. Ha, H. N. Phan, T. T. Thai, and N. M. Chung are with the School of Computer Science and Engineering, International University, Vietnam National University, Ho Chi Minh City, Vietnam.

L. H. Pham is with the College of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

Corresponding e-mail: hvusynh@hcmiu.edu.vn

978-1-7281-3679-0/19/$31.00 ©2019 IEEE

identified simultaneously. At the last step, our work uses visual features which are derived from MobileNets [10] conjugate a CNN model inspired from a Single Shot Multibox Detector [11] for the task of detecting obscured vehicles.

The remaining of this paper proceeds as follows: Initially, Section II describes our proposed method including Vehicle detection, Vehicle classification and Occlusion handling. Then, experiments and discussion are stated in Section III to evaluate, to encapsulate our work, and to conclude the paper.

II. The Proposed Method

A. Moving Vehicle Detection

Our method follows a motion-based approach where sequences of images are captured from static pole-mounted cameras. Then, a model of background subtraction is typically adopted to construct two main components: a background and a corresponding foreground from an input image. While backgrounds depict the appearance of stationary objects, foregrounds characterize moving objects as white blobs on black scenes. In empirical investigation, a background image is not always immutably available because of a number of external factors including camera jitters, illumination change, flows of chaotic vehicles, and presence of motionless foreground objects. A noteworthy technique is to examine an entropy of a Gaussian mixture model and disorder removal framework to neglect the disorder frames. Hence, in order to address thorny issues in background modelling, an approach proposed by Ha *et al.* [12] is adopted in our work. Fig. 1(a)-(c) illustrates a process of foreground construction and background modelling

With the constructed background, moving foreground objects are extracted using structural analysis of border [13]. Pre-processing and filtering operations, which aim to remove shadow [14], and noises are executed to refine the vehicles' images. Then a configuration of an observation zone is applied in order to diminish off-center vehicles and to extract objects' contours inside the zone by using our prior work [15]. From these, we obtain the set of vehicle candidates separately as illustrated in Fig. 1(d)-(k).

B. Vehicle Classification

Literally, vanilla decision trees classify data mainly based on the correlation between the entropies of input features and expected outcomes; so they cannot serve as an alternative to neural networks due to trade-offs occurring in the sense of generalization and interpretability. On the contrary, recent research has proven the effectiveness of deep convolutional neural networks in classifying a complicated set of objects' images via their hidden layers. With respect to our approach, after vehicles are previously extracted, we propose a hierarchical model that use a representation of a neural network to train a decision tree on features of vehicles' shapes.

1) Feature Extraction: In order to reduce computational cost of each detection instance of a vehicle, a list of ten high-level measurement features that represent the characteristics of ith vehicle at kth frame are extracted as described in Table I. These features mostly present geometric characteristic of vehicles's appearance.

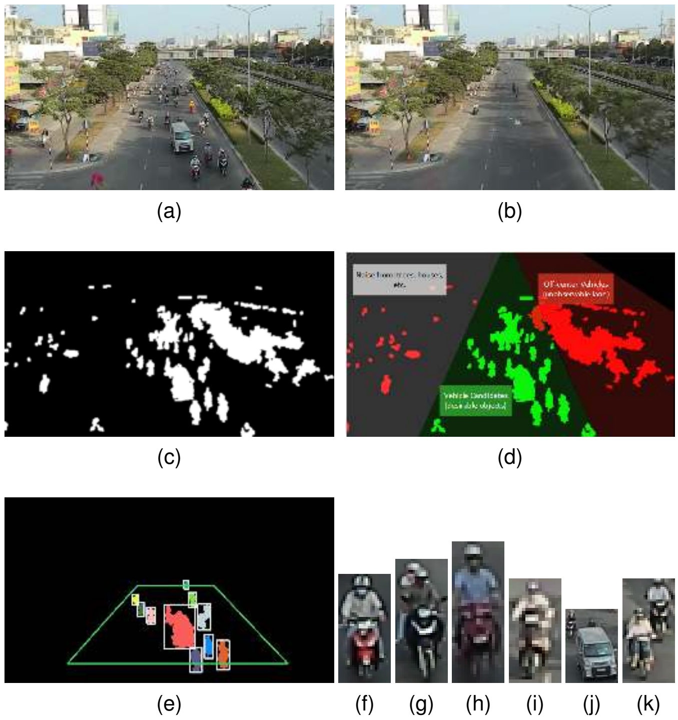

Fig. 1. Moving vehicle detection process. (a) Input image, (b) Constructed background, (c) Corresponding foreground, (d) Field of view of examining camera, (e) Detected vehicles within observation zone, (f) Some of extracted vehicles' images

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

2) Occlusion Detection: Among detected candidates, there are blobs of overlapping vehicles interspersed with isolated vehicles. From field of view of surveillance cameras, we regard vehicles' shapes as outward curves, which are illustrated in Fig. 3(a)-(f). Therefore, when inter-vehicle space reduces, defective areas surround vehicle areas as exemplified in Fig. 3(g)-(l). Accordingly, in order to separate these two group, we examine the defective area of each candidate i at each frame k, which is defined as:

$$P_i^D(k) = P_i^{CH}(k) - P_i^C(k) \tag{1}$$

Then, we investigate the occlusion state of each candidate by limiting this metric with a lower-bound threshold:

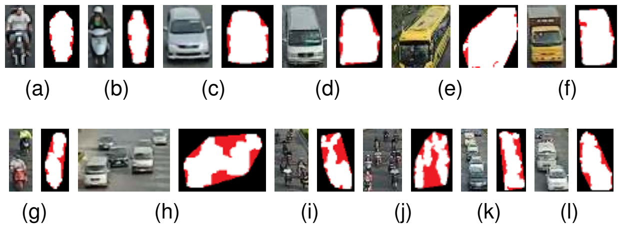

Fig. 3. Detected candidates with captured images and corresponding foreground. First row: isolated vehicles. Second row: blob of overlapping vehicles. White regions illustrate vehicles' areas. Red regions annotate defective areas.

$$\Phi_i^{occls}(k) = \begin{cases} 1 & \text{for } P_i^D(k) \geq \mathrm{T_{occls}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For each candidate, $\Phi_i^{occls} = 1$ indicates that object i presents blobs of overlapping vehicles. Otherwise, i is an isolated vehicle. Blobs of overlapping vehicles are further processed at a module of occlusion handling.

3) Neural Decision Tree model: Considering the variability of examining vehicles, we define three specific classes for the task of vehicle classification:

- **Class 1: Motorbike**, consisting of bike and motorbike.
- **Class 2: Car**, consisting of car, sedan, and SUVs.
- **Class 3: Large vehicle**, consisting of trucks and buses.

We construct a neural decision tree that utilizes the learning mechanism of neural networks to exploit common patterns of vehicles' features in form of a full binary tree as illustrated in Fig. 4(c). Our proposed decision tree contains a set of nodes that maintain two probabilistic attributes to route input features to leaf nodes for classification: a node probability and a branching probability. Overall, both of them are actually an analysis of conveyances' characteristics at each node; so these two attributes strongly depend on vehicles' features. Accordingly, using non-linear function is an appropriate approach to transform the features to pruning conditions. Regarding this issue, suppose a vehicle's feature X is being examined at an internal node $intl$ which has a left and a right node namely $intl_0$ and $intl_1$ respectively, the branching probability that $intl$ transfers X to a node $intl+1$ residing in the next lower level is presented as follows:

$$\begin{aligned} P_{intl}^{branch} &= P(intl+1|intl) \\ &= \begin{cases} \sigma(XW_i + B_i) & \text{for } (intl+1) = intl_0 \\ 1.0 - \sigma(XW_i + B_i) & \text{for } (intl+1) = intl_1 \end{cases} \end{aligned} \tag{3}$$

where σ is the sigmoid function; W_i, B_i respectively presents a learned weight and a bias of node $intl$

The other distribution that each node maintains is a node probability, denoted as P_i^{node}, which describes the probability that an examining set of features eventually reach a node i from the root node. Obviously, this metric follows the Bayes' theorem of conditional probability [16]. Furthermore, each feature vector incontrovertibly pass by the root node at the beginning. Mathematically, let $i-1$, $i-2$, $i-3$, ..., 2, 1, $root$ be sequential bottom-up ancestors of i, the probability that a vehicle feature reaches node i is equivalent to:

$$\begin{aligned} P_i^{node} &= 1.0 & &\text{if } i = root \\ P_i^{node} &= P_{i-1}^{node}.P_{i-1}^{branch} = P_{i-1}^{node}.P(i|i-1) & &\text{otherwise} \\ &= P_{root}^{node}.\prod_{j=root}^{i-1} P\left(j+1|j\right) \\ &= P_{root}^{node}.\prod_{j=root}^{i-1} P_j^{branch} \end{aligned} \tag{4}$$

There is a radical difference between leaf nodes and other internal nodes. Specifically, while internal nodes focus on directing vehicles' features to a node at the next level, leaf nodes play a key role in classifying vehicle into defined classes, labelling candidates of vehicles at the end of the tree. Different from the traditional paradigm of neural networks, at the lowest level of our tree, each leaf has a potential to perform compartmentalization for multiple classes. Given that X is examined at a leaf l, the probability which vehicle is classified in class c mathematically indicated as:

$$P_{l,c}^{cls} = P(c|l) = \frac{\exp(\phi_{l,c})}{\sum_{c' \in C} \exp(\phi_{l,c'})} \tag{5}$$

where $\phi_{l,c}$ denotes the learned parameter that leaf l distributes vehicle instances into class c of C classes.

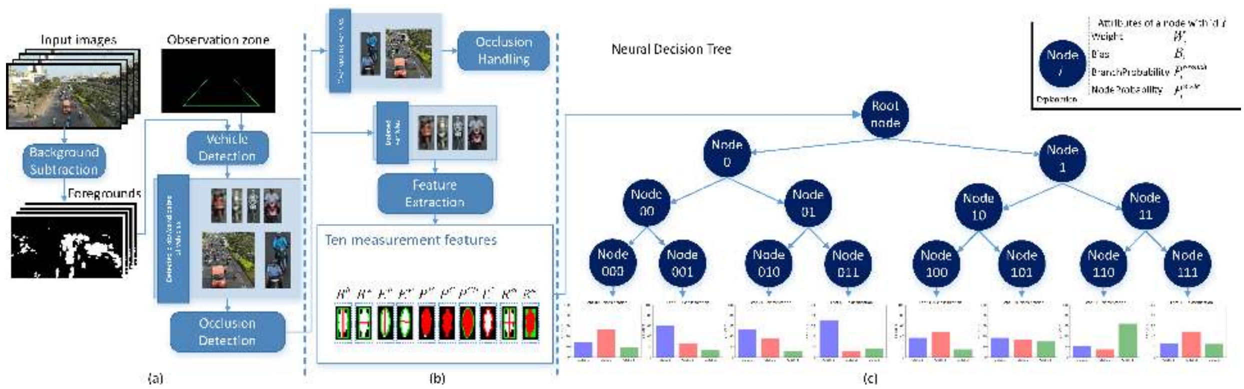

Fig. 4. Vehicle classification process. (a) Vehicle Detection. (b) Feature Extraction. (c) Vehicle Classification via a neural decision tree with $depth = 3$.

4) Class prediction: Turning into classification, when a set of vehicles' features is passed through the tree mentioned in the above section, there are two tasks that need to be taken on to determine corresponding categories of vehicles: choosing a leaf that performs classification for each vehicle and then figuring out a class of the vehicle based on the attributes of that leaf.

As previously stated in Section II-B3, during the phase of classification, the path routing of each vehicle is influenced by the characteristics of feature vectors. In other words, vehicles may have different choices of leaves to label their categories based on probabilistic metrics. For each vehicle, taking a leaf that possess the greatest node probability, we obtain a leaf handling the classification for that vehicle. Then, the class of a vehicle v that was categorized by a leaf l_v is defined as:

$$c_v = \underset{c \in C}{\operatorname{argmax}}\, P^{cls}_{l_v,c}(X_v) \tag{6}$$

$$\text{with} \quad l_v = \underset{l \in leaves}{\operatorname{argmax}}\, P^{node}_{l}(X_v) \tag{7}$$

5) Loss function: After defining a model for classification, a loss function is a significant factor that ensures expected outcomes. Regarding our approach, the loss function of the proposed neural decision tree consists of two factors: routing and classification.

For the perspective of routing vehicles through internal nodes, in order to enhance the utilization of all paths within the tree,we must make equal use of all nodes and conserve a balance between two branches of each node. Otherwise, some branches in the tree becomes inequitable and unreachable. Considering this issue, we construct an averaging factor for each node by following the principal idea of arithmetic mean. Literally, the probability of reaching a sub-branch of each node counts on the sum of the node probabilities of that node with respect to different sets of vehicles' features X:

$$\gamma_i = \frac{\sum_X P^{node}_i(X).P^{branch}_i(X)}{\sum_X P^{node}_i(X)} \tag{8}$$

In the literature, we hypothesize that two branches of each node have equal distribution of 0.5. Furthermore, the probability of reaching a node also proportionally depends on the depth of the node. With defined terms, the balance between two branches of each node is achieved by using entropy loss. Hence, the penalty for all internal nodes is defined as follows:

$$L_{route}(X) = -\lambda \sum_{i \in Intl.Nodes} \rho^{d_i} \left[0.5 \log(1.0 - \gamma_i) + 0.5 \log(\gamma_i)\right] \tag{9}$$

where d_i is the depth of node i in the tree, ρ is a decay penalty; λ is a regularization factor. both of these values are added to avoid overfitting in the model during training phase.

Another significant aspect of the proposed model is the loss of classification. Regarding this, we use cross-entropy as a cost function measuring the distance between two probability distributions - predicted and actual classification, denoted as:

$$L_{cls}(X) = -\log\left(\sum_{l \in leaves} \sum_{c \in C} \bar{Y}_c \log P^{pred}_{l,c}(X)\right) \tag{10}$$

$$\text{with} \quad P^{pred}_{l,c} = P^{node}_{l}(X) P^{cls}_{l,c}(X) \tag{11}$$

where $P^{pred}_{l,c}$ is the probability that a feature vector X is categorized as class c after going through the tree from the root and terminating at leaf l. The value of this metric is derived by following the Bayes's theorem of conditional probability.

In summary, the overall loss function of our proposed model is the sum of both loss functions. Trained with a sufficient dataset of corresponding vehicle features, the model updates elementary attributes at each node. Extracted features are then utilized to categorize isolated vehicles into three classes.

C. Occlusion Handling

Moving at a high density of the road area, vehicles become more challenging to be detected as their inner-space distance sharply decreases, increasing the number of overlapping vehicles. Regarding this issue, investigating vehicles' textured features is a pragmatic approach. In other words, the representation of vehicles are inspected throughout intermediate layers of a convolutional neural network. As extremely growing in theory, the demand for low-end devices is coming up as a huge need in deep learning. For recent years, there have been multiple approaches to reduce the gap between theory and practical use in various perspectives. However, only a few can achieve a real-time performance. With respect to our work, the main goal is to detect and classify types of vehicles, so only the networks dedicated for object detection stay in our focus. For this purpose, we used a model of the Single Shot MultiBox Detector (SSD) [11] with a backbone network of MobileNets [10]. By splitting input into channels and convolving separate filters for each channel, MobileNets achieves a significant improvement in classification. Furthermore, that advancement

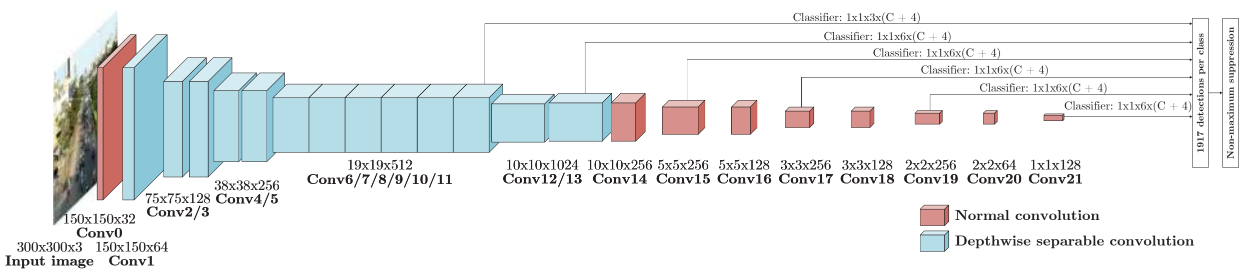

Fig. 5. Architecture of a convolutional neural network for occlusion handling

makes SSD outperform most of other networks with the combination of multibox detectors which presents multiple-scale visual features at in a single network

In this research, we retrain the SSD-MobileNets model in self-generated video datasets. During training procedure, the input image size is scaled down from a default size of 512×512 to 300×300 pixel as illustrated in Fig. 5. However, there is a slight modification when the model is adopted for our module of vehicle occlusion handling. Particularly, because we detected blobs of obscured vehicles in prior steps, we do not have to use a large size for input images. Also, SSD-MobileNets has a capability of processing multi-scale presentation of objects. Hence, when we integrate this model into our proposed method, images of overlapping vehicles are resized into 65×65 pixel in the detection phase. By using a smaller size for vehicles, our method reduces a great amount of calculation in the model.

III. Experiments and Discussion

For our problem, we use real-world video datasets previously collected on Pham Van Dong and Vo Van Kiet street, in Ho Chi Minh City, Vietnam. These cameras are installed at a height of $8-9$ meter with an inclined angle of $12^o - 15^o$ to the horizontal direction as presented in [15]. Our dataset consists of several 30 fps, 640×360 videos; each is of 10 minutes long. The traffic scenes consist of multiple classes of objects: pedestrian, motorbikes, 4-6 seat cars, small and large trucks, buses. Attributes of three examined datasets namely PVD01, PVD02, VVK01 are presented in Table II. In this method, we only work on vehicles on defined types as mentioned in Section II-B3. The proposed method has been implemented on a configuration of Intel Core i5, 16GB of RAM. Our method is implemented with Tensorflow and OpenCV on Python.

TABLE II
Characteristics of three experimented datasets

Dataset	Traffic Class	Traffic Amount	Frames	Scenario	Weather	Lane type
PVD01	1 2 3	1413 18 9	21,236	morning	overcast, sunny	mixed lanes for bikes and cars
PVD02	2 3	872 17	14,975	mid-afternoon	overcast	dedicated lanes for cars
VVK01	1 2	1001 65	16,298	mid-morning	shadow	mixed lanes for bikes and cars

A. Training

We conducted two parts of training with support of a GPU. First, for vehicle classification, we extracted a total of 14,013 vehicles' images from 3 video datasets: PVD01, PVD02, and VVK01. Ten corresponding features of these candidates are then calculated and labelled. The proposed decision tree that is proposed in this research is implemented with $\mathrm{T_{occls}} = 150$, a depth $d = 6$, a decay penalty $\rho = 0.9$, and a regularization factor $\lambda = 10$. Then, our proposed decision tree is trained on this set of data using Adam Stochastic Optimization with initial learning rate 0.01, and batch size 16. The model has been trained for three days with exactly 750 iterations.

Second, for occlusion handling, we labelled 7,919 vehicles from 650 images that are extracted from 3 video datasets. The network of SSD-MobileNets has been trained for a week with exactly one million iterations.

B. Overall Evaluation

In this research, we focus on evaluating two metrics: detection accuracy and processing time. In order to evaluate the detection accuracy, we compare the number of in-class objects predicted by three models (a traditional approach proposed by Phan *et al.* [17], a fine-tunning model of a vanilla SSD-MobileNets model [10], [11], and our proposed method) and calculate the class-wise recall rate and precision rate using the theory by Sokolova *et all.* [18]. For the processing speed, we perform benchmark real-time performance of examined methods without GPUs.

Table III, IV, and V show empirical results obtained from the analysis of three methods on different datasets. It is apparent that our proposed method is much more balanced in terms of both accuracy and performance than the others. It does not only achieves performance ratings that are on par with that of Phan's approach [17], but our method's accuracy ratings are also higher than those of SSD-MobileNets. With an average of 26.13 fps in real-time performance, the performance of our method is approximate 26.91 fps of Phan's method, superior to SSD-MobileNets's average of 15.88 fps. In addition, most of our precision-recall ratings are higher than those of SSD-MobileNets. On the VVK01 dataset, in terms of precision and recall, our predictions for Class 1 (the most populated class) achieved the higher respective scores of 89.67% and 91.91%, whilst SSD-MobileNet only came with respective ratings of 83.13% and 90.61%. For that same dataset, but with Class 2, our scores are at 81.69% and 89.23%, which are 2.28% and 6.15% higher than those of the aforementioned method. This result is achieved through reducing a great computational

TABLE III
Comparison of results on dataset PVD01

Method	Class	TP	TN	FP	FN	Precision	Recall	Perf.
Phan *et al.* [17]	1 2 3	856 11 6	17 862 867	716 9 5	557 7 3	54.45% 55.00% 54.55%	60.58% 61.11% 66.67%	27.46 fps
SSD-MobileNet [10], [11]	1 2 3	1271 15 8	23 1279 1286	163 4 2	142 3 1	88.63% 78.95% 80.00%	89.95% 83.33% 88.89%	16.05 fps
Proposed Method	1 2 3	1296 16 7	23 1303 1312	126 4 3	117 2 2	91.14% 80.00% 70.00%	91.72% 88.89% 77.78%	26.93 fps

TABLE IV
Comparison of results on dataset PVD02

Method	Class	TP	TN	FP	FN	Precision	Recall	Perf.
Phan *et al.* [17]	2 3	563 10	10 563	384 8	309 7	59.45% 55.56%	64.56% 58.82%	26.43 fps
SSD-MobileNet [10], [11]	2 3	753 14	14 753	124 4	119 3	85.86% 77.78%	86.35% 82.35%	15.67 fps
Proposed Method	2 3	790 15	15 790	107 4	82 2	88.07% 78.95%	90.60% 88.24%	24.98 fps

TABLE V
Comparison of results on dataset VVK01

Method	Class	TP	TN	FP	FN	Precision	Recall	Perf.
Phan *et al.* [17]	1 2	683 40	40 683	543 34	318 25	55.71% 54.05%	68.23% 61.54%	26.85 fps
SSD-MobileNet [10], [11]	1 2	907 54	54 907	184 14	94 11	83.13% 79.41%	90.61% 83.08%	15.92 fps
Proposed Method	1 2	920 58	58 920	106 13	81 7	89.67% 81.69%	91.91% 89.23%	26.47 fps

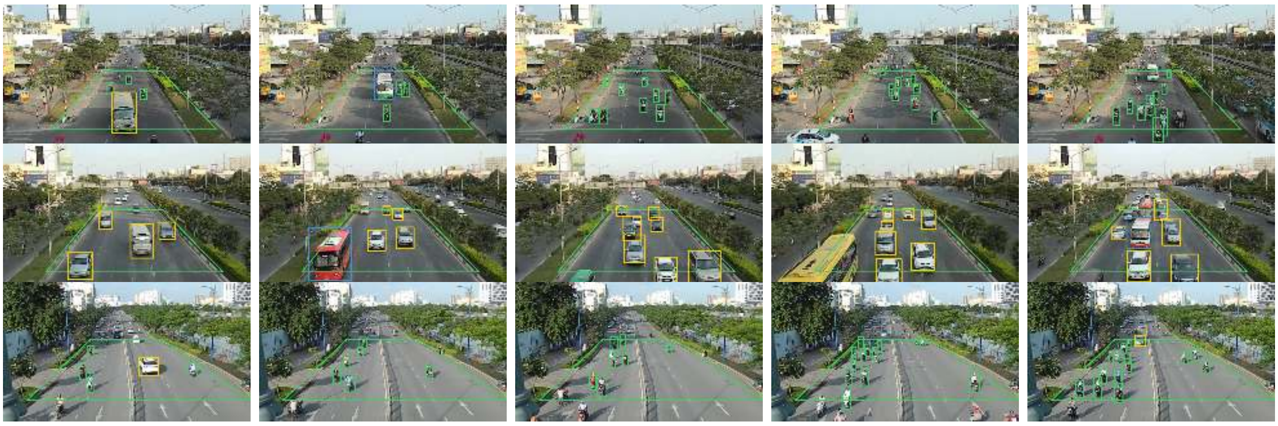

Fig. 6. Classification results with three experimental datasets with an increase of traffic density (from left to right). Green, yellow, blue rectangles respectively indicate vehicles in class 1, 2, 3. First row: dataset PVD01. Second row: dataset PVD02. Third row: dataset VVK01

amount on single-pipeline convolutional neural networks by narrowing down examining regions with background subtraction and transferring isolated vehicle classification to neural decision tree. However, the accuracy ratings for Class 3 in the PVD01 dataset are pronouncedly the lowest recorded accuracy duo in our prediction. However, in some minor situations, noise regions that present in traffic scenes form unexpected objects and slabs of foreground faults due to a variety of external factors including walking or sleeping foreground objects, illumination change, pixel camouflage. Fig. 6 presents some examples of three conducted experiments.

IV. Conclusion

In this paper, we proposed a novel method for detecting vehicles in daytime traffic scenes with a two-fold solution. First, we construct a neural decision tree to classify isolated vehicles. Next, obscured candidates are investigated on a fine-tuning SSD-MobileNets model with a smaller resolution of input image than the default structure. In this research, conducted experiments show that our proposed method is not only robust in multi-class vehicle classification with an average recall rate of 88.81% but also achieve a real-time performance. This scheme is capable of demonstrating a vision-based traffic surveillance system on unexceptional computer configuration when labelled training datasets are limited.

References

[1] F. Garcia, D. Martin, A. de la Escalera, and J. M. Armingol, "Sensor fusion methodology for vehicle detection," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 123–133, Spring 2017.

[2] D. N.-N. Tran, L. H. Pham, H. M. Tran, and S. V.-U. Ha, "Probabilistic model and neural network for scene classification in traffic surveillance system," in *Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing*, vol. 672, 2018, pp. 685–695.

[3] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and networked vehicle surveillance in its: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 25–48, Jan 2017.

[4] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3194–3200.

[5] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1827–1836.

[6] X. Ma and X. Sun, "Detection and segmentation of occluded vehicles based on symmetry analysis," in *2017 4th International Conference on Systems and Informatics (ICSAI)*, Nov 2017, pp. 745–749.

[7] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Vision-based occlusion handling and vehicle classification for traffic surveillance systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 80–92, Summer 2018.

[8] J. E. Espinosa, S. A. Velastin, and J. W. Branch, "Motorcycle detection and classification in urban scenarios using a model based on faster R-CNN," *CoRR*, vol. abs/1808.02299, 2018.

[9] N. Frosst and G. E. Hinton, "Distilling a neural network into a soft decision tree," *CoRR*, vol. 1711.09784, 2017.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. 1704.04861, 2017.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[12] 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.

[13] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32 – 46, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0734189X85900167

[14] L. H. Pham, H. N. Phan, D. H. Le, and S. Viet-Uyen Ha, "A hybrid shadow removal algorithm for vehicle classification in traffic surveillance system," in *Intelligent Engineering Informatics*, V. Bhateja, C. A. Coello Coello, S. C. Satapathy, and P. K. Pattnaik, Eds. Singapore: Springer Singapore, 2018, pp. 647–655.

[15] S. V.-U. Ha, L. H. Pham, H. M. Tran, and P. H. Thanh, "Improved vehicles detection and classification algorithm for traffic surveillance system," *Journal of Information Assurance and Security*, vol. 9, no. 5, pp. 268–277, 2014.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006, ch. Probability Distributions.

[17] H. N. Phan, L. H. Pham, D. N.-N. Tran, and S. V.-U. Ha, "Occlusion vehicle detection algorithm in crowded scene for traffic surveillance system," *2017 International Conference on System Science and Engineering (ICSSE)*, pp. 215–220, 2017.

[18] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427 – 437, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457309000259