# Real-Time Illegal Parking Detection Algorithm in Urban Environments

Xinggan Peng, Rongzihan Song, *Graduate Student Member, IEEE*, Qi Cao, Yue Li, Dongshun Cui, Xiaofan Jia, *Graduate Student Member, IEEE*, Zhiping Lin, *Senior Member, IEEE*, and Guang-Bin Huang *Senior Member, IEEE*,

***Abstract*—Currently, illegal parking detection tasks are mainly achieved through manually checking by enforcement officers on patrol or using Closed-Circuit Television (CCTV) cameras. However, these methods either need high human labour costs or demand installation costs and procedures. Therefore, illegal parking detection solutions, which can reduce significant labour and equipment installation costs, are highly demanded. This paper proposes a novel voting based detection algorithm using deep learning networks implemented using in-vehicle cameras to achieve illegal parking detection with multiple offences' types. Adopting in-vehicle cameras better matches real-world mobile scenarios than using traditional CCTV cameras as this helps enforcement authorities to reduce manpower and installation costs. A well-constructed new dataset with more than 10,000 high-quality labelled images with seven object categories is built for illegal parking detection tasks. Additionally, one novel labelling method named "minimal illegal units" is proposed for illegal parking detection. It reduces the time and human labelling costs significantly, achieving a better correlation of a vehicle and its parking type. The experiments have been conducted in the urban areas of Singapore. Furthermore, the illumination robustness test has also been performed to illustrate that the proposed detection algorithm exhibits strong resistance to changing illumination conditions in varied operating environments. Our proposed detection algorithm can provide a benchmark for research in illegal parking detection.**

***Index Terms*—Illegal parking detection, in-vehicle camera, deep learning neural network, multi-class classification.**

## I. Introduction

THE problem of "illegal parking" has been a significant concern that draws public attention because of the rapidly increasing number of vehicles in contrast with limited parking resources in cities [1], [2], [3], [4], [5], [6]. Illegal parking activities include parking vehicles in restricted regions or parking in unauthorized manners. Illegal parking activities not only cause congestion to the traffic flow but also lead to common civil complaints for urban management [7]. In order to impede the occurrence of illegal parking activities, government authorities usually deploy enforcement officers to conduct manual patrols. Some countries also deploy Closed-Circuit Television (CCTV) cameras in certain regions to detect illegal vehicle parking. For example, Urban Redevelopment Authority (URA) of Singapore manages about 700 roadside parking sites with about 17,000 parking lots in Singapore. Currently, the enforcement checking is manually performed by patrolling officers, who have to manually sight each of illegal parking activities, take photos and record offences. However, these methods are expensive with high labour costs and inefficient to cover large areas. Therefore, illegal parking detection solutions are highly demanded, which can reduce manpower and equipment installation costs significantly. Such detection solutions can be further exploited for intelligent transportation systems (ITS).

Recently, many efforts have been made to detect illegal parking activities using CCTV cameras. Both Akhawaji $et\ al.$ [8] and Sarker $et\ al.$ [9] apply an image segmentation algorithm based on Gaussian Mixture Model to extract vehicle information. Akhawaji $et\ al.$ [8] further improve the vehicle tracking by using Kalman Filter to reduce false alarms. However, the performance of this method may suffer if rapid changes in illumination of operating environments occur. Another improved foreground extraction method applies one-dimensional transformation projection to input images, which reduces the dimension and computational complexities of the extraction process [10]. Deep learning methods have also been applied to illegal parking detection. For example, Convolutional Neural Networks (CNN) and Single Shot MultiBox Detector (SSD) networks have been implemented into illegal parking detection CCTV systems [11], [12], [13]. Compared with traditional image segmentation algorithms, deep learning networks are more robust to the change of nearby operating environments such as illumination and weather conditions. However, since CCTV cameras are stationary with presetting the region of interest (ROI), the detection coverage and scenarios are limited. For example, Ng $et\ al.$ [11] use a stationary IP camera to detect parked cars at an outdoor illegal parking lot. The work reported in [12] can detect illegal parking along the roadside, but without further classified illegal parking types. There are three illegal parking detection public datasets using CCTV, i.e., Imagery Library for Intelligent Detection Systems (i-LIDS) [8], dataset provided by ISLab [14] and Sussex

Manuscript received *xxxxxx*; revised *xxxxxx*; accepted *xxxxxx*. This work was supported by the National Research Foundation, Singapore through its AI Singapore Programme, under AI Singapore Award, under Grant AISG-100E-2018-014 and by National Research Foundation Singapore, Competitive Research Programme, under Grant NRF-CRP18-2017-02. *(Corresponding author: Qi Cao.)*

Xinggan Peng, Rongzihan Song, Xiaofan Jia, and Zhiping Lin are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

Qi Cao is with School of Computing Science, University of Glasgow, Singapore 567739 (email: qi.cao@glasgow.ac.uk).

Yue Li was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. He is now with I Innovations Private Ltd., Singapore 318995.

Dongshun Cui and Guang-Bin Huang were with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. They are now with Mind PointEye, Singapore 608526.

Traffic Monitoring dataset of the University of Sussex [15].

Nowadays, in-vehicle cameras are widely installed in most vehicles as they are very useful vehicle accessories. Compared to stationary CCTV cameras, in-vehicle cameras have several advantages in ITS applications, such as no limitation to fixed physical locations, low installation requirements and low capital expenditure [16]. Applications of in-vehicle cameras include driver assistance systems [17] in rainy [18] and foggy weather [19], traffic sign detections [20], [21], [22], [23], etc. Matsuda $et\ al.$ [24] present an algorithm to detect all vehicles parked along streets using dashboard camera videos without further distinguishing parking areas or road line types. However, in practice, not all vehicles parked along roadsides are illegal if there are not restricted signs or restricted road lines. It is hence necessary to relate a vehicle and its corresponding parking area or the type of road lines to make accurate illegal parking detections. To the best of our knowledge, the detection of illegal parking activities with detailed types of offence using in-vehicle cameras has not been reported in the literature.

In this paper, a novel voting based real-time illegal parking detection algorithm is developed using videos captured by in-vehicle cameras. The detection algorithm is implemented using deep learning networks to detect parking types of vehicles in real-time. The developed algorithm has great advantages not only in easy installation and low cost, but also in high resistance to the change of nearby environments such as low illumination or rainy weather. Besides, prior knowledge of the ROI is not required for the proposed algorithm, hence achieving high quality and high accuracy detection performance for multiple illegal parking activities in different operating scenarios. Additionally, a new dataset is built by our research team, for dealing with illegal parking detection tasks using in-vehicle cameras. Moreover, one novel labelling method named "minimal illegal units" is also developed during the dataset construction, which significantly reduces the labour and time cost of labelling procedures. It better relates the vehicle to its detailed parking type. Compared with conventional labelling methods, which focus on specific margins between objects, the developed labelling method marks vehicles' information and offence factors mutually without marking individual margins.

The main contributions of this paper are as follows.

1) The proposed novel voting based algorithm firstly achieves illegal parking detection with multiple types of detailed offence using in-vehicle cameras with benchmark results. To the best of our knowledge, our proposed algorithm is the first research work to achieve such functionalities, since this area has not been exploited in prior studies.

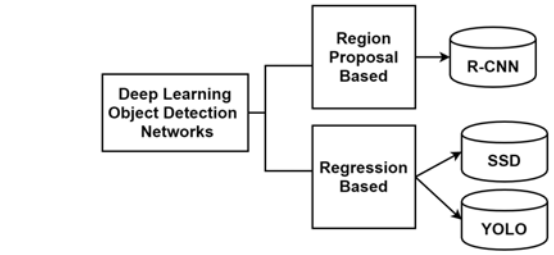
2) The structure of the developed algorithm is straightforward to add or fine-tune the sub-networks to improve the overall performance for future research.

3) A newly constructed illegal parking detection dataset contains more than 10,000 high quality labelled images with seven object categories, including six common illegal parking types and one legal parking type. It is the first dataset using in-vehicle cameras to deal with illegal parking detection without relying on fixed-location ROIs and it further classifies captured vehicles into seven detailed types with ground truth labels.

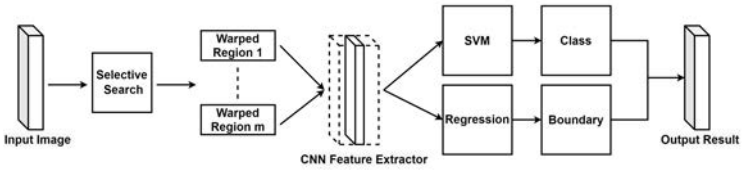
4) A novel labelling method named "minimal illegal units" for illegal parking detection is proposed. This labelling method efficiently links the vehicle and essential parking information to achieve detailed parking types classification.

As a limitation, at the current stage of the research work, the data are all collected in Singapore. Roadside parking activities at many locations are treated as illegal, where some are not allowed for parking at all time; others are not allowed for parking during the daytime, but permitted for parking at night. As such, the new dataset is imbalanced: the number of data samples collected from daytime is more than its counterpart at night.

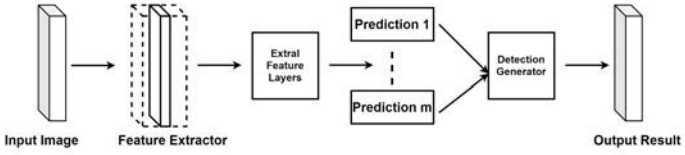
The remaining parts of this paper are organized as follows. Section II reviews the background and related works in the literature. Section III presents the procedures and standards of data collection in our research. Section IV describes the details of the new dataset and the real-time illegal parking detection algorithm. Section V presents the experimental design for benchmark results. Section VI concludes the paper.



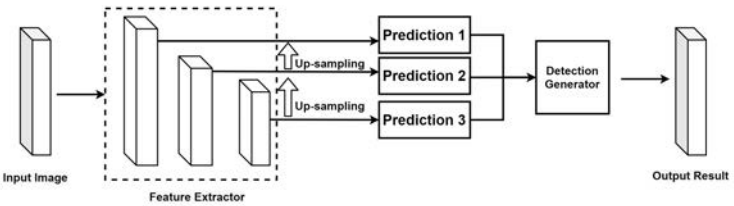

(a)




(b)




(c)




(d)

Fig. 1: (a) Two types of deep object detection networks: region proposal based and regression based. (b) Flowchart of R-CNN (c) Flowchart of SSD (d) Flowchart of YOLOv3.

## II. Background and Related Works

Nowadays, many artificial intelligence (AI), machine learning and deep learning based methods have been applied in various applications such as edge computing [25], elderly healthcare [26], vehicular networks [27] and object detection, etc. The goal of illegal parking detections is to identify parking offences and further determine their locations in images. Therefore, these tasks have close relations to the scopes of object detection. Typically, object detection tasks can be handled using general machine learning or deep learning methods.

According to [28], object detection models generally have three modules: region selection, feature extraction, and object classification. Traditional machine learning based object detection pipeline contains two main modules: ROI extraction and object classification [29]. One typical ROI extraction method applies sliding windows with different scales shifting all over an image [30]. However, due to enormous computational complexities, exhaustive sliding is very difficult to be applied in practice. Lowe $et\ al.$ [31] have made improvements in processing speed of pedestrian detection by assuming geometry symmetry of people, predefining aspect ratio and applying the stereo-vision technique to refine distance information. Instead of using exhaustive sliding windows, Enzweiler $et\ al.$ [32] introduce a keypoint selection method by finding maximum and minimum values among different Gaussian functions with the ratio of principal curvatures.

For object classification tasks, several discriminative classification methods aim to find optimal decision boundaries between different candidate classes after feature extraction [33]. Primarily, feed-forward multilayer neural networks [34] apply linear discrimination functions to non-linearly map samples. Support Vector Machines (SVM) [35] with kernels linearly map samples in higher feature space. Dalal and Triggs [36] introduce a method which applies linear SVMs combined with Histogram of Orientation (HOG) features to reduce false-positive rate in human detection tasks. Freund $et\ al.$ [37] introduce another method to perform classification task by cascading AdaBoost classifiers [38].

The prosperity of deep learning networks and performance improvements benefit from multiple factors [28], [39], such as large training data sets [40], [41], powerful graphics processing unit (GPU), and novel training strategies [42]. Because of distinguished learning and adaptive capacities, deep learning networks have been widely applied to object detection tasks, which can be categorized into two types, as shown in Fig. 1(a). One type begins with region proposals which are then further classified into backgrounds or classes, such as R-CNN [43]. The other type focuses on treating the detection task as a regression problem and eventually building one-stage networks such as SSD [44] and You Only Look Once (YOLO) [45].

As shown in Fig. 1(b), the model of R-CNN firstly determines the number of bounding box object region candidates using selective search. Then features are computed by a large CNN network. Later classes and boundaries of targets are predicted by binary SVMs trained for each class and linear regression, separately. Although R-CNN can achieve satisfactory performances in some object detection tasks, the training time of such models is very long, and the detection speed is limited [46].

In order to increase both training and detection speed, one-stage networks such as SSD and YOLO have been introduced to deal with object detection tasks. As shown in Fig. 1(c), the region based proposals are replaced by directly applying CNN based feature extractors on the whole image. Then extra convolutional feature layers at the end of the feature extractor will generate predictions at multiple scales by convolutional filters. Due to the less complex feature extractor without using region based proposals, the detection speed of SSD has increased a lot compared with two-stage networks. A flowchart of another example of a one-stage network, YOLO, is shown in Fig. 1(d). Instead of treating class prediction and boundary locations separately, the model of YOLO directly generates detection proposals on both boundaries and class probabilities based on the feature map produced by a feature extractor. It consists of multiple convolutional layers with filters of size of $1 \times 1$ and $3 \times 3$. YOLO treats an object detection task as a regression problem to reduce the detection time. An input image will be divided into $m \times m$ grid cells. Each cell will generate $n$ bounding boxes with five factors (i.e., centre coordinates: $x, y$, width: $w$, height: $h$ and object confidence scores for each bounding box). The object confidence is defined as $Pr(Object) \times IOU_{pred}^{truth}$, where $Pr(Object)$ indicates the probability of the existence of an object; $IOU_{pred}^{truth}$ represents the Intersection over Union (IOU) overlap area between predicted results and ground truth. Meanwhile, every cell generates $p$ conditional class probabilities $Pr(Class_j|Object)$. During the test, class-specific confidence scores for every box can be computed in Eq. (1).

$$\begin{aligned} Pr(Class_j) \times IOU_{pred}^{truth} =& Pr(Class_j|Object) \\ & \times Pr(Object) \times IOU_{pred}^{truth} \end{aligned} \tag{1}$$

The final layer of YOLO generates both classification probabilities and coordinates of corresponding bounding boxes, which are normalized to be in the range of 0 to 1. In total, output of each grid cell is encoded as a tensor with the size of $m \times m \times (n \times 5 + numbr\ of\ Classes)$. Except for the last layer, which uses a linear activation function, all other layers use the leaky rectified linear activation function to increase non-linearity as shown in Eq. (2).

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{Otherwise} \end{cases} \tag{2}$$

An improved version YOLOv3 applies a more powerful feature extractor with more convolutional layers and residual layers to achieve better detection performances. Furthermore, YOLOv3 predicts boxes at three different scales. To be specific, three feature maps of different scales are obtained. Two of them are generated by transmitting the original feature map using twice up-sampling on previous layers, which is similar to procedures in SSD. This strategy assists the network to obtain more semantic information from the less-sampled features and the earlier feature map. Batch normalization is

applied to stabilize and increase the speed of the training process. YOLOv3 has significant benefits over other state-of-the-art methods on detection speed and has been applied in many real-world applications [46], [47], [48], [49].

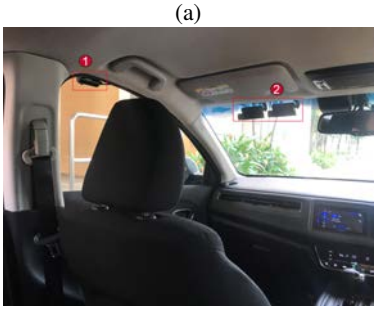
## III. DATA COLLECTION

In this section, data collection apparatus and collection procedures of the new dataset are introduced using in-vehicle cameras for illegal parking detection.

### A. Collection Apparatus

The data collection is conducted using a Honda Vezel car shown in Fig. 2(a) with cameras being installed, which start recording videos when driving. This type of car model is very common in Singapore. The installation positions of the in-vehicle cameras are preferred being at high altitude for better reviews during video recording. Fig. 2(b) shows the layout of the apparatus related to recording. There are three horizontally placed 1080p-Full-HD cameras, which can record videos with up to 30-frame per second. Item No. 1 is a side-view camera to support front-view cameras to locate vehicles and parking lots. Item No. 2 includes two front-view cameras, but only one camera is used to detect and recognize vehicles with parking sites. The other camera is set as the backup.

(a)




(b)

Fig. 2: Experiment vehicle and apparatus used in the data collection. (a) Experiment vehicle (b) apparatus layout. Item No. 1 is the side-view camera. Item No. 2 are the two front-view cameras with only one being used for detection.

### B. Collection Tasks and Procedures

The primary task is to collect videos of roadside parking vehicles with as many different scenarios as possible, such as different traffic conditions, street blocks, weather, and illumination conditions. The experiment vehicle is driven by an experienced driver to keep the speed maintaining around 40 Km/h, which is the average driving speed of the enforcement vehicles in urban areas of Singapore. In order to collect data samples under different illumination conditions and traffic situations, the collection tasks are performed mainly once a week within three time blocks: 9 am - 12 pm, 2 pm - 5 pm, and 7 pm - 9 pm during June 2019 to January 2020. Loop tours among the candidate roads, such as Veerasamy Road, Desker Road, Gul Way, and Aliwal Street, etc., are conducted for data collection tasks with the in-vehicle cameras. These candidate roads are selected according to the suggestions of Singapore government authority experts, where illegal parking activities are likely to happen. This strategy allows the data collection team to have sufficient recording-time gaps in between for the same candidate spot. It further increases the probability of capturing different parking vehicles and different types of illegal parking, as some of illegal parking activities are rare, such as double parking. Consequently, after finishing preliminary data processing, a few additional trips are made to candidate spots, where rare cases are likely to happen making our data less biased. In total, around 50-hours-long video footages have been collected.

## IV. Dataset Details and Real-Time Illegal Parking Detection Algorithm

### A. Dataset Details and Statistics

Object detection tasks aim to identify the existence of the objects as well as their locations in images or videos [50]. Many object detection tasks have been applied to ITS. However, object detection in real traffic situations is challenging because a large number of candidate objects vary from shapes, colours, and dimensions, with different environment conditions, such as illumination and weather, etc. [51]. To be specific, detection of illegal parking activities needs to overcome the effects of similar vehicles with slight differences in their parking details and other background effects such as pedestrians or objects on roadsides. Compared with CCTV, the main advantages of in-vehicle cameras include no limit in locations and ROI. Therefore, the attributes of the new dataset enable the detection algorithm to specify illegal parking offence details. Taking Singapore as an example, there are mainly six types of illegal parking activities listed in TABLE I. As such, samples in the new dataset are classified into seven parking types including six illegal types and one legal type.

After collecting raw video footages, a detailed data processing is performed with the flowchart shown in Fig. 3. Firstly, since illegal parking activities do not regularly exist through constant recording, footages are required to be manually clipped into short videos that contain illegal parking activities. Secondly, manually labelling the learning contents frame by frame from the selected videos is required, including vehicle types, vehicle margins, and “minimal illegal units”. Vehicle types and vehicle margins are used to generate pre-detected vehicle regions. While the “minimal illegal units” are used for the illegal units detection module to detect types of illegal parking offence.

Every vehicle margin in each frame at the pixel-level is labelled by combining with vehicle types and parking types,

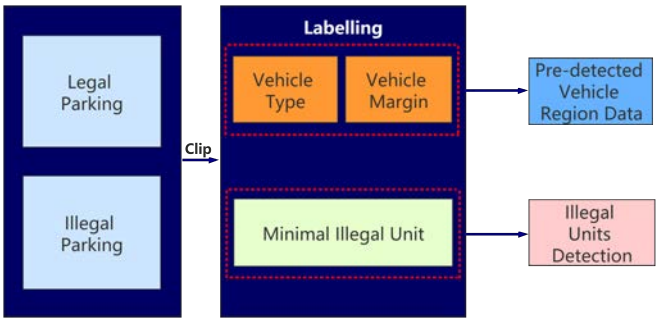



Fig. 3: Data processing flowchart including raw footages clipping and image labeling.

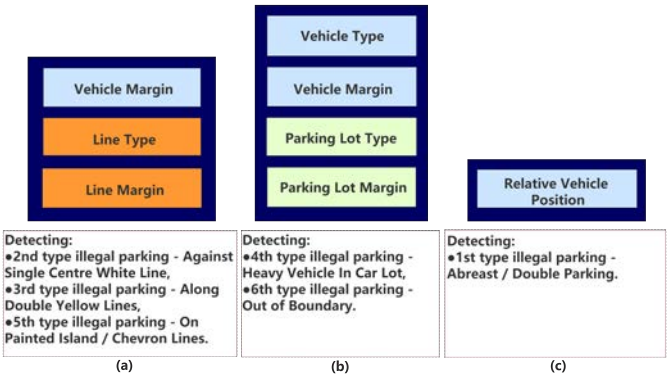



Fig. 4: Information required for detecting different types of illegal parking. (a) detecting the $2^{nd}$, $3^{rd}$ and $5^{th}$ illegal parking types; (b) for detecting the $4^{th}$ and $6^{th}$ illegal parking types; (c) for detecting the $1^{st}$ illegal parking type.

TABLE I: Predefined Label Tags

| No | Vehicle Type | No | Parking Type (one legal and six illegal types) |
|---|---|---|---|
| 1 | Car | 1 | $1^{st}$ illegal parking type: Abreast / Double Parking |
| 2 | Heavy Vehicle | 2 | $2^{nd}$ illegal parking type: Against Single Centre White Line |
| 3 | Motorcycle | 3 | $3^{rd}$ illegal parking type: Along Double Yellow Lines |
| | | 4 | $4^{th}$ illegal parking type: Heavy Vehicle In Car Lot |
| | | 5 | $5^{th}$ illegal parking type: On Painted Island / Chevron Lines |
| | | 6 | $6^{th}$ illegal parking type: Out of Boundary |
| | | 7 | Legal parking type: Legal |

in order to generate pre-detected vehicle regions, which will be used in the overall algorithm construction. As such, its label tag is derived accordingly, such as "Car: Along Double Yellow Line", "Heavy Vehicle: Against Single Centre White Line", etc. Note that heavy vehicles also include truck, bus, etc. The details of predefined vehicle types and parking status tags are listed in TABLE I.

Besides vehicle regions, detecting specific types of illegal parking activities requires combining different essential information shown in Fig. 4. Conventional image labelling methods that require label objects separately in pixel-level will cost a lot of human efforts and capitation. If the illegal parking activities belong to Fig. 4(a) and (b), conventional image labelling methods need to firstly mark the exact margins of the lines and parking lots as well as the vehicles separately. Next, corresponding label tags are required to be assigned to different margins. If the activities belong to Fig. 4(c), conventional image labelling methods will become more difficult and inappropriate, as more complex margins of target objects need to be separated. For example, if the illegal parking activities belong to "Double parking", conventional labelling methods need to separately mark nearby vehicles, which may require polygon boxes rather than rectangle boxes. It increases significantly the labelling difficulty. Even if the vehicle locations and line margins have been precisely labelled, it is hard to determine the location relationship between the line margin and the vehicle location among separate labelling boxes. Such confusion may result in false alarms.

In order to address such problems, a novel labelling method named "minimal illegal units" is introduced in this paper. It is able to reduce the complex labelling workload for labelling illegal parking objects and efficiently link the vehicle's location with essential parking information. In general, the definition of "minimal illegal units" refers to the minimum area acceptable

Fig. 5: Illustration sample of "minimal illegal units". The rectangle in red is the acceptable "minimal illegal units" and rectangles in orange are other defective units.

for humans to determine vehicle parts and illegal parking offence details. One sample illustration of "minimal illegal units" is shown in Fig. 5. In this figure, two rectangles in orange are either too small or too large, such that either illegal offence details are missed, or unnecessary background information is contained. On the contrary, the rectangle in red in the acceptable region contains both vehicle parts and illegal parking offence details with a minimal area size for detection tasks. Compared with conventional labelling methods, which emphasize specific boundaries between objects, the proposed labelling method requires less time and manpower. It allows to combine multiple types of information into one rectangular box without separating their boundaries. Both the vehicle's location and illegal details are extracted simultaneously within a region which contains little unwanted background information. Therefore, both individual characteristics and the location relationship of these two components can be obtained for the illegal parking detection module. Nevertheless, to maintain top learning performances, during the image labelling, on the one hand, vehicle parts inside the labelling box should not be too small to be regarded as a vehicle part. On the other hand, the proportion of vehicle parts and illegal factors such as a segment of yellow or a corner of one parking lot should not vary significantly for all labelled samples.

Using the proposed "minimal illegal units" labelling method, images are labelled according to the six types of illegal parking. Fig. 6 shows labelling examples according to

TABLE II: Number of Samples in the Constructed Dataset for Each Illegal Parking Type

| Illegal Parking Type | Number of Samples |
|---|---|
| $1^{st}$ illegal parking type: Abreast / Double Parking | 1702 |
| $2^{nd}$ illegal parking type: Against Single Centre White Line | 3580 |
| $3^{rd}$ illegal parking type: Along Double Yellow Lines | 4859 |
| $4^{th}$ illegal parking type: Heavy Vehicle In Car Lot | 269 |
| $5^{th}$ illegal parking type: On Painted Island / Chevron Lines | 1693 |
| $6^{th}$ illegal parking type: Out of Boundary | 1818 |

the required information mentioned in Fig. 4. To be specific, for illegal parking activities belonging to Fig. 4(a) such as "$2^{nd}$ illegal parking type: Against Single Centre White Line", "$3^{rd}$ illegal parking type: Along double yellow line" and "$5^{th}$ illegal parking type: On Painted Island / Chevron Lines", the "minimal illegal units" are set as the area that contains the wheel, which can represent the vehicle information and parts of the road line. After testing and fine-tuning the detection performance, the proportion between the area of vehicle and road line is set to be around 2:1.

Illegal parking activities within types of Fig. 4(b), such as "$4^{th}$ illegal parking type: Heavy Vehicle In Car Lot" and "$6^{th}$ illegal parking type: Out of Boundary", parts of the vehicle (including the front or the rear wheel) with the parking lot's corner being labelled as the "minimal illegal units".

Similar way is applied to Fig. 4(c) on "$1^{st}$ illegal parking type: Abreast / Double Parking", where parts of the head or back of the illegally parked vehicle and the partially blocked vehicle are marked as "minimal illegal units".

In total, there are 13,921 labelled illegal parking samples in 10,972 images in the constructed dataset by our research team. These samples are further classified into six illegal parking types. The number of collected samples for each type of illegal parking is shown in TABLE II.

### *B. Comparison with Three Relevant Public Datasets*

The new dataset aims to include the details that are not covered in the three mentioned public datasets based on CCTV. This new dataset contains large number of illegally parked vehicles and recorded street blocks with different illegal parking types' details. Sufficient samples and recorded street blocks cover most of the practical scenarios that enforcement authority may encounter.

Furthermore, since the new dataset has non-fixed ROIs for each frame, it can better prevent the algorithm from stacking in the non-object features to generate overfitting phenomenon [39] during the training. A larger number of samples can assist the proposed detection algorithm to achieve better generalization. The comparison of the new dataset and these mentioned CCTV based ones is shown in TABLE III.

### *C. Real-Time Illegal Parking Detection Algorithm*

Given an input image obtained from the collected videos, the task is to determine whether or not vehicles in this frame are parked legally or illegally with specific offences' types. Fig. 7 illustrates the architecture of the proposed algorithm, which consists of two modules: offline training and real-time detection. The training data is used to achieve offline training of the illegal units detection and to determine the IOU overlap area threshold $\tau$. In addition, in the real-time detection pipeline, the trained illegal units detection module processes the input image to generate illegal units detection outputs. Then the result is combined with the corresponding vehicles' regions data based on IOU overlap area threshold. It is followed by the voting based process that is applied to generate output parking types of vehicles. Each of these components is explained in the following Section IV-D and Section IV-E.

### *D. Offline Training*

YOLOv3 [52] has come out with a more reliable trade-off between accuracy and detection speed than earlier versions. YOLOv3 makes some changes in its detection strategies such as using logistic loss and predicting boxes with three different scales. As a result, YOLOv3 can manage many real-time object detection tasks with relatively high processing performances.

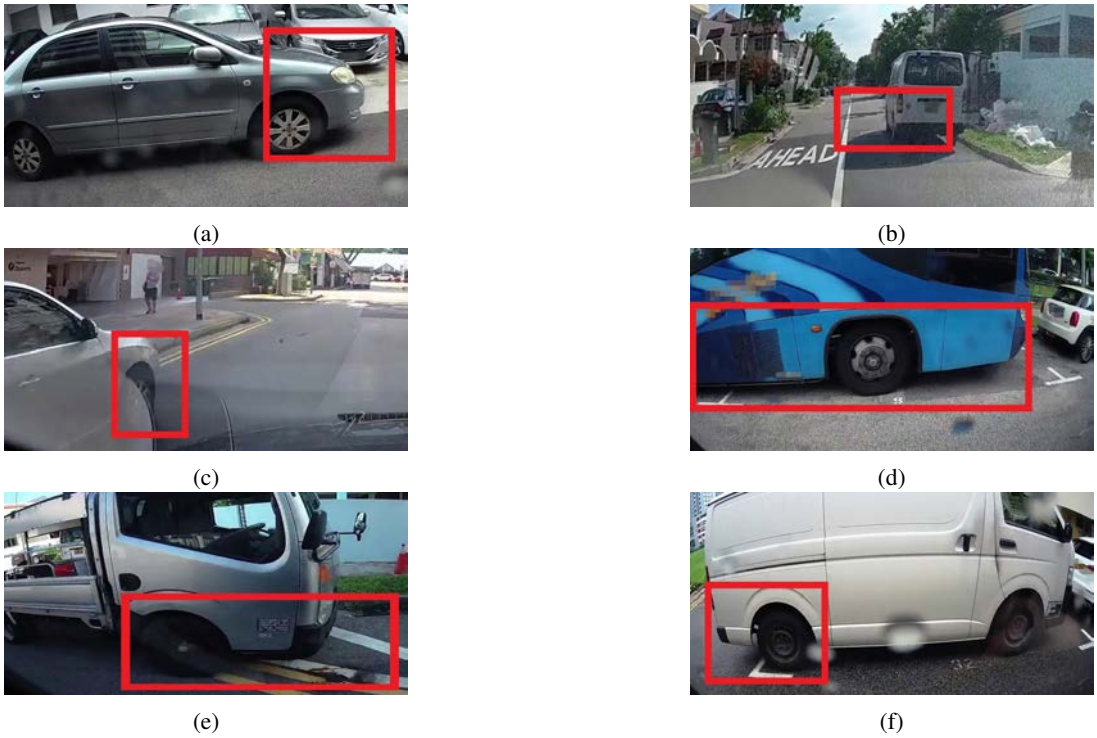
One YOLOv3 deep learning network is used for illegal units detection in Fig. 7. There are two types of data used for training: labelled vehicle region training data and labelled illegal units training data. The training set for the vehicle region contains data $\{X_1, T_1\} = \{x_{1i}, t_{1i}\}_{i=1}^{n}$, where $x_{1i}$ represents the coordinates of rectangular labelled vehicles and $t_{1i}$ represents the corresponding vehicle types defined in TABLE I.

Similarly, the training set for the illegal units detection module contains data $\{X_2, T_2\} = \{x_{2i}, t_{2i}\}_{i=1}^{m}$, where $x_{2i}$ contains coordinates of the rectangular labelled illegal units and $t_{2i}$ is the corresponding illegal parking types, excluding the type "Legal" defined in TABLE I.

The minimum IOU overlap area threshold $\tau$ among the vehicle regions and detected illegal units is determined based on the IOU overlap value between $\{X_1, T_1\}$ and $\{X_2, T_2\}$.

For the detection module, mini-batches with a batch size of eight are used to divide corresponding training data. The network performs forward propagation on the current mini-batch to compute their output and loss. Next, stochastic gradient descent (SGD) [53] method is used to achieve weights update during back propagation. After parameters tuning and evaluations, for the networks of the illegal units detection module, the default learning rate is set to be $\eta_0 = 0.001$ and gradually decreases when the train process is close to finishing the total batches [54]. The momentum and decay of both networks are $\varphi = 0.9$ and $\lambda = 0.0005$, respectively.

In the training process, YOLOv3 model aims to optimize the multi-task loss function [46], [52] shown in Eq. (3).



(a) (b) (c) (d) (e) (f)

Fig. 6: Labelling examples of six types of illegal parking activities: red rectangles are training examples of the "minimal illegal units". (a) the $1^{st}$ illegal type: "Abreast / Double Parking". (b) the $2^{nd}$ illegal parking type: "Against Single Centre White Line". (c) the $3^{rd}$ illegal parking type: "Along Double Yellow Lines". (d) the $4^{th}$ illegal parking type: "Heavy Vehicle In Car Lot". (e) the $5^{th}$ illegal parking type: "On Painted Island / Chevron Lines". (f) the $6^{th}$ illegal parking type: "Out of Boundary".

TABLE III: Comparisons of the New Dataset with Common CCTV Illegal Parking Detection Datasets

| Dataset Content | **The New Dataset** | i-LIDS [8] | ISLab-PVD [14] | Sussex Traffic Monitoring [15] |
|---|---|---|---|---|
| Type of Camera | In-Vehicle Camera | CCTV | CCTV | CCTV |
| Daytime / Night | Both | Both | Both | Both |
| Multiple Objects | ✓ | ✓ | ✓ | ✓ |
| Non-fixed ROI | ✓ | × | × | × |
| Number of Illegal Parking Sample Category | **6** | 1 | 1 | 1 |
| Number of Recorded Street Block | **>30** | N.A. | ≈16 | N.A. |

$$
\begin{aligned}
&Loss = \\
&\alpha_{coord}\sum_{i=0}^{m^2}\sum_{j=0}^{n}\mathbb{I}_{ij}^{obj}[(x_i-\widehat{x}_i)^2+(y_i-\widehat{y}_i)^2] \\
&+\alpha_{coord}\sum_{i=0}^{m^2}\sum_{j=0}^{n}\mathbb{I}_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\widehat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\widehat{h}_i}\right)^2\right] \\
&+L_{Con}+L_{Class}
\end{aligned}
\tag{3}
$$

where the first and the second lines are the coordinates loss of the bounding boxes; $\widehat{x}_i, \widehat{y}_i, \widehat{w}_i, \widehat{h}_i$ are ground truth of the coordinates; and $\mathbb{I}_{ij}^{obj} = 1$ if there exist objects in $j^{th}$ bounding box derived by $i^{th}$ cell, otherwise $\mathbb{I}_{ij}^{obj} = 0$.

Notations of $L_{Con}$ and $L_{Class}$ are Confidence loss and Classification loss as shown in Eq. (4) and Eq. (5), respectively.

$$
\begin{aligned}
L_{Con} = &-\sum_{i=0}^{m^2}\sum_{j=0}^{n}\mathbb{I}_{ij}^{obj}[\widehat{S}_i log(S_i)+(1-\widehat{S}_i)log(1-S_i)] \\
&-\alpha_{noobj}\sum_{i=0}^{m^2}\sum_{j=0}^{n}\mathbb{I}_{ij}^{noobj}[\widehat{S}_i log(S_i)+(1-\widehat{S}_i)log(1-S_i)]
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
L_{Class} = -\sum_{i=0}^{m^2}\mathbb{I}_{i}^{obj}\sum_{p_i\in classes}&[\widehat{Pr}(p_i)log(Pr(p_i)) \\
&+(1-\widehat{Pr}(p_i))log(1-Pr(p_i))]
\end{aligned}
\tag{5}
$$

where $\mathbb{I}_{ij}^{noobj}$ is the complement of $\mathbb{I}_{ij}^{obj}$; $S_i$ is the confidence score; $Pr(p_i)$ is the probability of class $p_i$; $\widehat{S}_i$ and $\widehat{Pr}(p_i)$ are the ground truth of the confidence score and probability of class $p_i$, respectively.

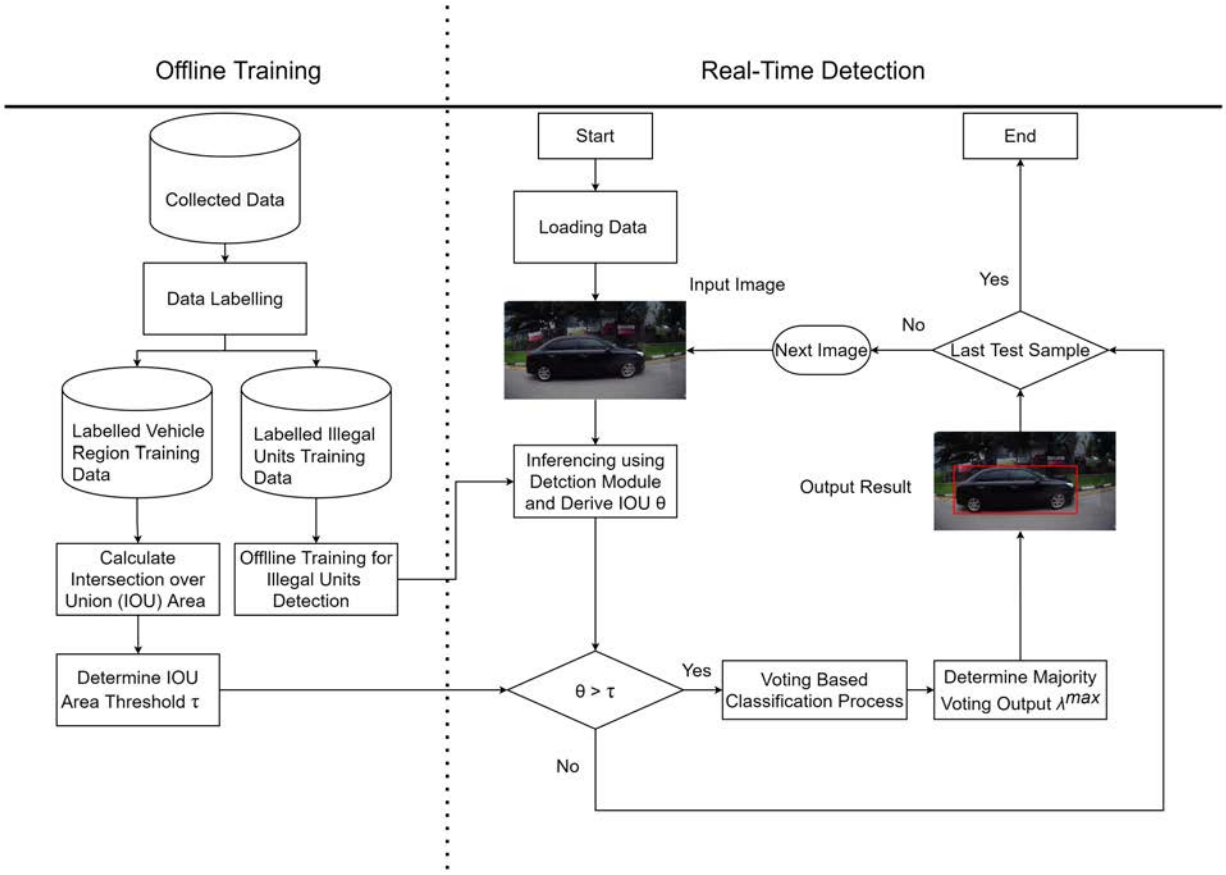



Fig. 7: Flow chart of the proposed detection algorithm.

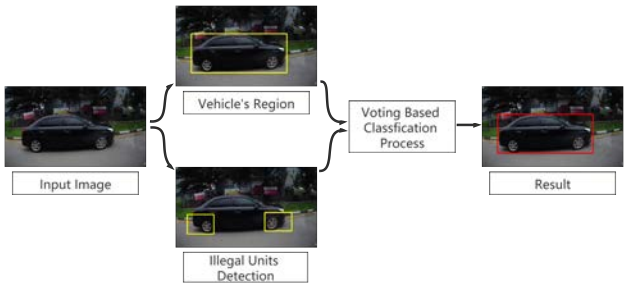



Fig. 8: Illustrated example of the refined detection strategy.

Sigmoid activation function is used to produce $S_i$ and $p_i$. But one drawback is that the output of the sigmoid function derivative will become too small if its input is large. As a result, the model's convergence speed becomes very slow if we use a loss function like mean squared error (MSE), which was applied on YOLOv1, making the error value too small. To overcome such drawback, the cross-entropy loss function is applied to $L_{Con}$ and $L_{Class}$ in Eq. (4) and (5) to reduce the negative effect of the small values generated by the sigmoid function derivative during the backpropagation [55]. Different detection tasks have different sample characteristics. For example, if most grid cells of samples from one task do not contain an object, the influence of scarce cells containing objects may need to be increased to make the model focusing more on the cells that contain objects. Therefore, $\alpha_{coord}$ in Eq. (3) and $\alpha_{noobj}$ in Eq. (4) are hyper-parameters to balance the loss from cells containing objects and non-object. To be specific, if $\alpha_{coord}$ is set to be higher than $\alpha_{noobj}$, it puts more emphasis on cells containing objects and leads to better recall performance.

Evaluations of the hyper-parameter setting of the detection module will be discussed in Section V. Since the experimental vehicle may make turns frequently during the practical use, both front-view and side-view images are combined to train one network to make it less sensitive to change of view. Moreover, in order to make the algorithm more robust to the change of illuminations, artificial data augmentations by changing values of brightness, saturation and contrast within the statistical results of the collected data are applied to the training data during the training.

### *E. Real-time Detection*

The detection goal is to determine the parking type of each vehicle rather than to detect locations of human-defined "minimal illegal units" ground truth boxes, subject to manual deviations. Slight location shifts between such predicted units and ground truth units do not necessarily affect the parking type of one vehicle. Therefore, the illegal parking detection can be treated as a multi-class classification problem of parking

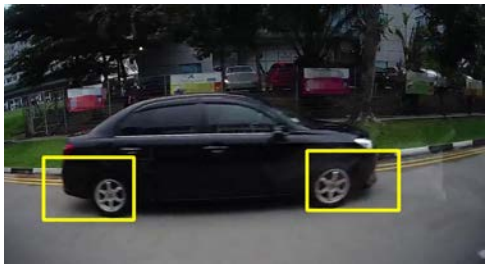


Fig. 9: Example of single vehicle with multiple "minimal illegal units".

types for vehicles that appear in each frame, rather than a pure object detection problem for "minimal illegal units". The illustrated procedures of the refined detection strategy are shown in Fig. 8. This strategy makes the algorithm focusing on the determination of the parking type of each vehicle. It is our goal to alleviate the impacts of small location deviations from the predicted "minimal illegal units". Such small deviations are not significant for the results of vehicle parking type classification. Besides, one vehicle may have multiple "minimal illegal units", an example being shown in Fig. 9. To get appropriate output parking type, a majority voting based process [56] is adopted in the developed algorithm.

Each input image $s$ of the test dataset $S$ is processed by the illegal units detection module to generate $I$ detection outputs as shown in Eq. (6).

$$O_i^{IP} = (x_i^{IP}, y_i^{IP}, w_i^{IP}, h_i^{IP}, s_i^q, \lambda_i^q); i = 1, ..., I \quad (6)$$

where $x_i^{IP}, y_i^{IP}, w_i^{IP}, h_i^{IP}$ are boxes locations values; $s_i^q$ and $\lambda_i^q$ represent the confidence score and the detected illegal parking type among number of $Q$ $(Q = 6)$ illegal parking types for the $i^{th}$ detection box, respectively. Then every predicted output $O_i^{IP}$ from the illegal units detection module will be assigned based on the IOU overlap value to $N$ vehicles' regions in each image frame as shown in Eq. (7).

$$O_j^V = (x_j^V, y_j^V, w_j^V, h_j^V); j = 1, ..., N \quad (7)$$

where $x_j^V, y_j^V, w_j^V, h_j^V$ are vehicle's locations values.

The minimum threshold $\tau$ is set to filter out the output $O_i^{IP}$ with too small IOU values $\theta_i$. So the $j^{th}$ vehicles' region $O_j^V$ contains number of $K$ illegal parking type $\lambda_i^q$ with corresponding confidence score $s_i^q$. The final output parking type for each $O_j^V$ is then determined by majority voting on number of $K$ assigned illegal parking type $\lambda_k^q$ and corresponding confidence score $s_k^q$, where $k = 1...K$.

A vector $C_{O_j} \in \mathbf{R}^Q$ with the same number of target illegal parking types is used to record $K$ illegal units detection results that are assigned to the $j^{th}$ vehicle's region $O_j^V$. The value of the $q^{th}$ entry in the vector is the sum of the confidence score $s_k^q$ whose corresponding illegal parking type $\lambda_k^q$ is $q$, as shown in Eq. (8).

$$C_{O_j}(q) = \sum_{\lambda_k^q \in q} s_k^q \quad (8)$$

The parking type output of every vehicle's region $O_j^V$ is determined by a majority voting process, as shown in Eq. (9).

$$\lambda_j^{max} = \underset{q \in [1,...,Q]}{\arg\max} \{C_{O_j}(q)\} \quad (9)$$

If a vehicle's region $O_j^V$ has not been assigned with any illegal parking type, the output parking type of that vehicle will be "Legal".

Therefore, if one $O_j^V$ has been assigned with at least one $O_i^{IP}$, the output parking type is shown in Eq. (10).

$$O_j^{Final} = (x_j^V, y_j^V, w_j^V, h_j^V, \lambda_j^{max}); \quad j = 1, ..., N \quad (10)$$

where $\lambda_j^{max}$ is the output of the majority voting process for $O_j^V$. Otherwise, if there is no $O_i^{IP}$ being assigned to $O_j^V$, the results will be as shown in Eq. (11).

$$O_j^{Final} = (x_j^V, y_j^V, w_j^V, h_j^V, Legal); \quad j = 1, ..., N \quad (11)$$

The pseudocode of our real-time detection algorithm is presented in Algorithm 1.

**Algorithm 1:** Real-time Detection Algorithm

**Input:** Test Dataset $S$, number of illegal parking type $Q$ and IOU threshold $\tau$

```
for s ∈ S do
    Obtain I detection outputs with confidence score s_m^q from the trained detection module;
    for i ∈ I do
        IOU θ_i ← using eq. (6) and eq. (7);
        if θ_i > τ then
            for q ∈ Q do
                Vector C_{O_i}(q) ← s_i^q;
            end
        else
            C_{O_i}(q) ← 0
        end
    end
    λ^max ← arg max_{q∈[1,...,Q]} {C_{O_i}(q)};
    if λ^max ≠ 0 then
        Output parking type ← λ^max;
    else
        Output parking type ← Legal
    end
end
```

## V. Experiments

In this section, performance evaluation, benchmark and simulation results and the hyper-parameter setting for the real-time illegal parking detection algorithm developed in Section IV are presented. The ability of the developed algorithm to handle illumination changes is also analyzed. Additionally, detection performance comparisons and time cost analysis with other commonly used detection models are performed.

TABLE IV: Benchmark Performance Results for the Proposed Algorithm for Illegal Parking Detection

| Parking Type Number | Parking Type Name | Precision ( % ) | Recall ( % ) |
|---|---|---|---|
| 1 | Abreast / Double Parking | 88.73 | 91.95 |
| 2 | Against Single Centre White Line | 91.53 | 94.98 |
| 3 | Along Double Yellow Lines | 83.47 | 96.83 |
| 4 | Heavy Vehicle In Car Lot | 95.38 | 89.48 |
| 5 | On Painted Island / Chevron Lines | 95.45 | 94.12 |
| 6 | Out of Boundary | 80.39 | 77.78 |
| 7 | Legal | 96.69 | 92.50 |
| | | **Average Precision ( % )** | 90.24 |
| | | **Average Recall ( % )** | **91.09** |
| | | **F-Score ( % )** | **90.66** |

TABLE V: Comparisons Results with Relevant Detection Models for Illegal Parking Detection

| Parking Type Number | Parking Type Name | ResNet-50 [57] | | ResNet-101 [57] | | RetinaNet-50 [58] | | RetinaNet-101 [58] | | SSD[44] | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision ( % ) | Recall ( % ) | Precision ( % ) | Recall ( % ) | Precision ( % ) | Recall ( % ) | Precision ( % ) | Recall ( % ) | Precision ( % ) | Recall ( % ) |
| 1 | Abreast / Double Parking | 91.10 | 85.94 | 92.17 | 85.82 | 91.96 | 88.80 | 92.19 | 88.21 | 86.90 | 78.94 |
| 2 | Against Single Centre White Line | 98.04 | 78.11 | 98.84 | 74.20 | 94.98 | 70.27 | 92.69 | 70.07 | 97.36 | 70.06 |
| 3 | Along Double Yellow Lines | 86.87 | 82.22 | 90.37 | 82.27 | 83.22 | 83.35 | 84.72 | 83.82 | 78.98 | 75.53 |
| 4 | Heavy Vehicle In Car Lot | 87.76 | 96.06 | 89.23 | 94.91 | 77.28 | 95.28 | 76.84 | 96.85 | 89.42 | 50.30 |
| 5 | On Painted Island / Chevron Lines | 96.20 | 87.09 | 97.67 | 86.98 | 94.89 | 88.28 | 95.38 | 86.92 | 97.67 | 75.07 |
| 6 | Out of Boundary | 88.32 | 65.02 | 93.52 | 63.17 | 86.19 | 62.79 | 83.59 | 63.77 | 85.54 | 33.62 |
| 7 | Legal | 88.20 | 96.41 | 87.64 | 98.15 | 86.87 | 94.34 | 86.85 | 94.16 | 83.64 | 97.45 |
| **Average Precision ( % )** | | | 90.93 | | **92.78** | | 87.91 | | 87.47 | | 88.50 |
| **Average Recall ( % )** | | | 84.41 | | 83.64 | | 83.30 | | 83.40 | | 68.71 |
| **F-Score ( % )** | | | 87.54 | | 87.97 | | 85.55 | | 85.38 | | 77.36 |

### A. Experiment Setup

The experiments are conducted with Pytorch [59] deep learning framework which is run on a NVIDIA TESLA P100 GPU. The 3-fold cross-validation processes [60], [61] are conducted. To be specific, we split the entire collected dataset into three non-overlapping folds: one fold is reserved for testing, the other two folds are served for training. Each fold has nearly the equal number of data samples. Each of these three folds is used for testing once, whose performances are recorded. The average performance of these three tests is derived accordingly.

### B. Evaluation Metric

Since labels of parking types have already been assigned to each vehicle, the detection problem can be treated as the multi-class classification for each vehicle. The following five evaluation metrics shown in Eq. (12) to (16) are employed to evaluate the performances of the developed algorithm:

$$Precision(Class_j) = \frac{TP_j}{TP_j + FP_j} \tag{12}$$

$$Recall(Class_j) = \frac{TP_j}{TP_j + FN_j} \tag{13}$$

$$Precision = \frac{1}{M}\sum\nolimits_{j=1}^{M} \frac{TP_j}{TP_j + FP_j} \tag{14}$$

$$Recall = \frac{1}{M}\sum\nolimits_{j=1}^{M} \frac{TP_j}{TP_j + FN_j} \tag{15}$$

$$F\text{-}Score = (1+\gamma^2)\frac{Precision \times Recall}{\gamma^2 Precision + Recall} \tag{16}$$

where $TP_j$ is the number of $j^{th}$ type parking vehicles that are correctly classified; $FP_j$ is the number of NOT $j^{th}$ type parking vehicles that are misclassified as $j^{th}$ type; $FN_j$ is the number of $j^{th}$ type parking vehicles that are misclassified as NOT $j^{th}$ type; $M$ is the number of classes; $\gamma$ is the parameter to define the importance weight between *precision* and *recall*.

The computations of $TP_j$, $FP_j$, and $FN_j$ are shown in Eq. (17) to (19).

$$TP_j = \begin{cases} TP_j + 1, & \text{if } \lambda^{max} = j \text{ and } GT = j \\ 0, & \text{Otherwise} \end{cases} \tag{17}$$

$$FP_j = \begin{cases} FP_j + 1, & \text{if } \lambda^{max} = j \text{ and } GT \neq j \\ 0, & \text{Otherwise} \end{cases} \tag{18}$$

$$FN_j = \begin{cases} FN_j + 1, & \text{if } \lambda^{max} \neq j \text{ and } GT = j \\ 0, & \text{Otherwise} \end{cases} \tag{19}$$

where $GT$ is the ground truth label and $\lambda^{max}$ is the prediction of parking type produced by the proposed method shown in Eq. (10).

*Precision* and *recall* are commonly used in the evaluation of classification problems to represent the positive predictive rate and sensitivity of one algorithm. *F-Score* [62] combines *precision* and *recall* into a single metric by using a harmonic mean calculation. Here, because of the equal importance of *precision* and *recall*, we set $\gamma = 1$.

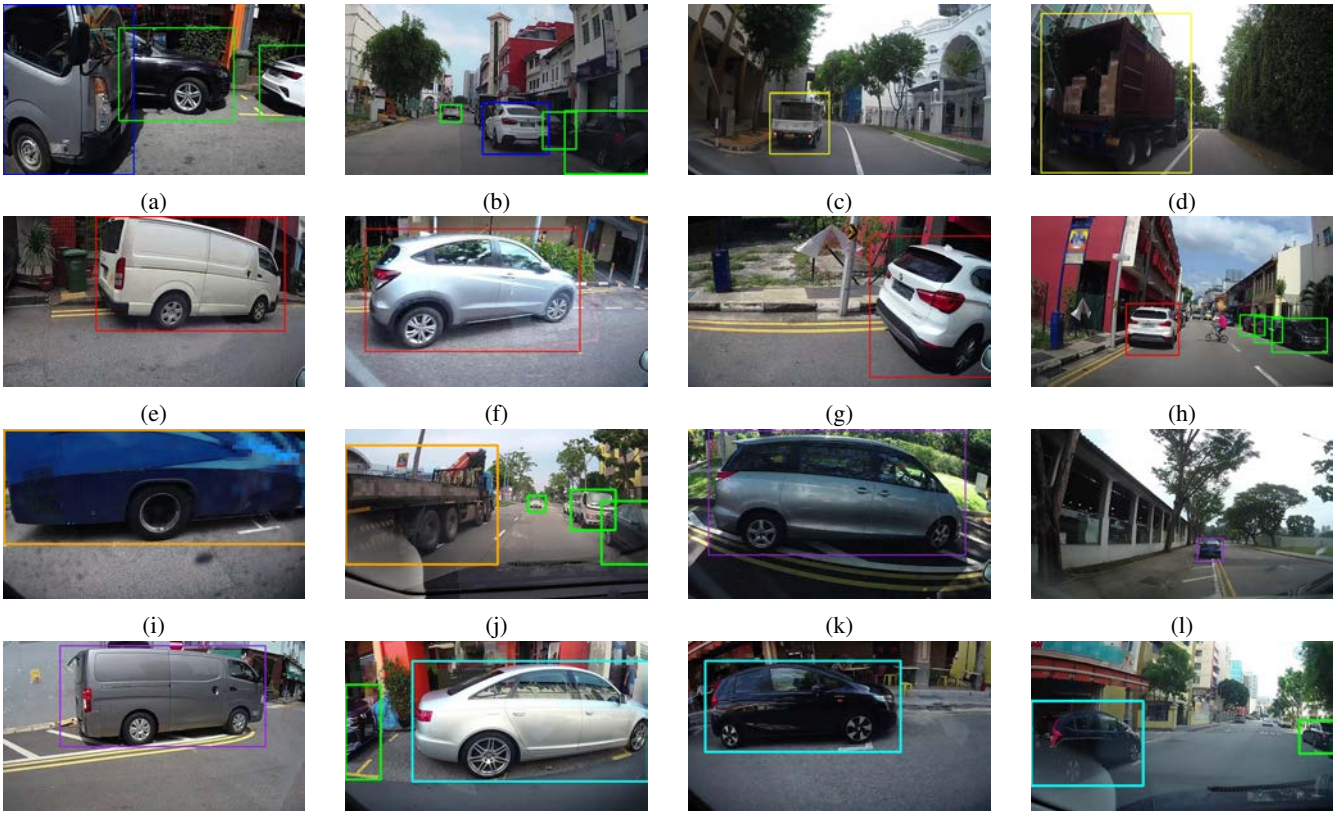



Fig. 10: Visualization example of illegal parking detection results. Every figure shows the front view or side view example for one of the six illegal parking types. Note that only front view images for $2^{nd}$ illegal parking type: "Against Single Centre White Line" are captured, because the side view camera is set on the left side. But the single centre white line always lays on the centre of the road lane, making it unable to be recorded by this camera.

### C. *Evaluation of Illegal Parking Detection Performance*

Even if there are multiple vehicles in one frame, the proposed algorithm can successfully detect the illegal parking activities and classify them within six illegal parking types or one legal parking type. To evaluate the ability of the proposed algorithm in preventing false alarm, *precision* and *recall* values of vehicles with the parking type "Legal" are also calculated in the experiments.

Fig. 10 shows detection examples for every parking type in the experiment. Each colour of the rectangular boxes represents one specific parking type as follows:

- Blue - $1^{st}$ illegal parking type: "Abreast / Double Parking",
- Yellow - $2^{nd}$ illegal parking type: "Against Single Centre White Line",
- Red - $3^{rd}$ illegal parking type: "Along Double Yellow Lines",
- Orange - $4^{th}$ illegal parking type: "Heavy Vehicle In Car Lot",
- Purple - $5^{th}$ illegal parking type: "On Painted Island / Chevron Lines",
- Cyan - $6^{th}$ illegal parking type: "Out of Boundary",
- Green - Legal type: "Legal".

TABLE IV presents the benchmark results tested on the constructed dataset. Data collected from the front-view and side-view in-vehicle cameras are evaluated together because mixed training method is applied in offline training. Observed in TABLE IV, the overall *F-Score* of the proposed algorithm is 90.66%. *Average Precision* for all seven parking types is 90.24% and *Average Recall* is 91.09%. The outcomes of these evaluation metrics show that the proposed algorithm is able to deal with illegal parking detection tasks efficiently. After analysing some mistaken cases such as those shown in Fig. 11, these cases are reasonably considered as very hard to be differentiated. To be specific, as shown in Fig. 11(a), 11(b) and 11(c), in some image frames, traffic lines such as "Single Centre White Line" or "Painted Island / Chevron Lines" or the parking lot conner are not clear, which makes it difficult to recognise them as illegal parking activities. Fig. 11(d) shows an example of very ambiguous case, where the head of the vehicle has parked out of the parking lots corner. This is the obvious characteristics for type "Out of Boundary" but since "double yellow lines" is near the vehicle, it is reasonable for type "Along Double Yellow Lines" also. As a result, under this circumstance, it is hard to make correct detection. Note that both values of precision and recall for type "Legal" are above or near 90%, which means the algorithm has a good

ability to reduce false alarms.

In addition, performance comparisons with related detection models are presented in TABLE V. Five other commonly used detection models including ResNet-50 [57], ResNet-101 [57], RetinaNet-50 [58], RetinaNet-101 [58] and SSD [44] are selected to replace the YOLOv3 detection models to compare the illegal parking detection performance. The first two models, i.e., ResNet-50 and ResNet-101, are based on the baseline Faster R-CNN [63] detection system with different backbone structures. RetinaNet-50 and RetinaNet-101 are two single unified network composed of ResNet-50 and ResNet-101 backbones with Feature Pyramid Network (FPN) structure [64], respectively. The backbone is applied to generate convolutional feature maps over an entire input image. SSD is a widely used recent one-stage detection network. Comparing the results presented in TABLE IV and TABLE V, results of *Average Precision* of ResNet-50 and ResNet-101 are 90.93% and 92.78%, respectively. While these two results are a little higher than the *Average Precision* of the proposed algorithm, their *F-Score* and *Average Recall* are both lower than that of the proposed algorithm. This shows that overall the proposed algorithm achieves satisfactory generalization and robustness.

Compared with performances produced by one-stage models including RetinaNet-50, RetinaNet-101 and SSD, all metrics including *Average Precision*, *Average Recall* and *F-Score* produced by the proposed algorithm with YOLOv3 as detection models are much higher than these detection models. Observed from the comparison results, the proposed algorithm can detect different illegal parking types efficiently.

In order to perform the time cost analysis, the offline training time cost and detection time cost of the proposed algorithm are evaluated and compared with other five models listed in TABLE VII. Their implementation environments are kept unchanged in the experiments to make it a fair comparison. As shown in TABLE VII, the proposed algorithm requires the minimum offline training time compared with the other five models. The ResNet-50 and ResNet-101 are two-stage based models that require using region proposal network (RPN) to generate region proposal, which are followed by the stage of classification and bounding box regression. As a result, these two models need much more time in training with 27.52 hours and 62.68 hours, respectively. Although RetinaNet-50 and RetinaNet-101 are one-stage networks, they have a rather large FPN structure backbone with two subnetworks. One is for classifying anchor boxes; and the other is for regressing anchors to target objects boxes. As such, their training processes still need long time with 9.86 hours and 21.24 hours, respectively. Moreover, SSD uses a large CNN based feature extractor in the network. It takes about 37.65 hours for training. On the contrary, the proposed algorithm adopts YOLOv3 as the detection model that directly deals with the bounding box regression problem on each grid cell. Therefore, the training speed is much faster, which is about 6.23 hours. Similarly, the YOLOv3 model has a much faster detection speed than the other models. As such, the proposed algorithm achieves the least detection time per image among these models, which is around 0.07 second. Other one-stage models take shorter detection time than that of the two-stage models (i.e., 0.21 and 0.24 second for RetinaNet-50 and RetinaNet-101, respectively and 0.17 second for SSD), but is still longer than that of the proposed algorithm.

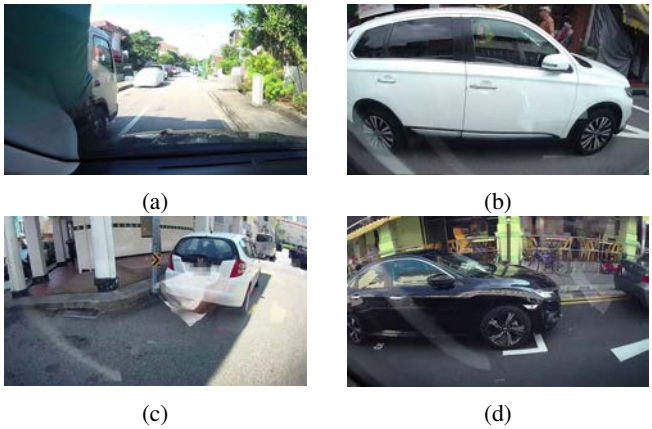


Fig. 11: Examples of mistakes during the experiment. (a), (b), and (c) are mistakes caused by unclear traffic lines or parking lot conner. (d) is the mistake caused by confusion of $3^{rd}$ illegal parking type: "Along Double Yellow Lines" and $6^{th}$ illegal parking type: "Out of Boundary".

Moreover, the time complexity of the proposed algorithm has also been evaluated and compared with those five models in terms of floating point operations (FLOPs). We assume that computational load of the proposed algorithm is mainly contributed by the deep learning networks in the detection module. The batch normalization and ReLU operations are negligible comparing to the operations by the convolution layers in the network architecture. As such, for each convolution layer, there are $(2 \times C_i \times k) \times HW \times C_o$ operations, where $C_i$ denotes the input channel; $k$ denotes the square of kernel size; $HW$ denotes the output feature map area; and $C_o$ denotes the output channel. Based on the analysis of the operations above, the complexity results are shown in the TABLE VII. It is observed that the proposed algorithm has much smaller number of FLOPs than those of these four models, ResNet-50, ResNet-101, RetinaNet-50, and RetinaNet-101, but slightly larger number of FLOPs than that of the SSD model. Note that the proposed algorithm has lower training time cost and detection time cost than that of SSD, and also better detection performances (e.g., F-Score: the proposed algorithm: 90.66%, SSD: 77.36%).

### *D. Hyper-parameter Setting*

YOLOv3 naturally handles multiple object detections and has few hyper-parameters to tune, which is more flexible for illegal parking detection. The learning rate is the hyper-parameter that controls the learning progress. Too large learning rate leads to unwanted diverge, while too small learning rate seriously slows down the learning process. Optimizers link together the model weights and the loss function by updating the model in response to a decrease in the output of the loss function. Several optimizers are commonly used in training deep neural networks such as SGD and Adam [65]. However, the choice of optimizers normally depends on the

TABLE VI: Performance of Illumination Robustness for the Proposed Algorithm of Illegal Parking Detection

| Parking Type Number | Parking Type | Precision ( % ) | Recall ( % ) |
|---|---|---|---|
| 1 | Abreast / Double Parking | 88.65 | 91.60 |
| 2 | Against Single Centre White Line | 91.45 | 94.95 |
| 3 | Along Double Yellow Lines | 82.70 | 96.71 |
| 4 | Heavy Vehicle In Car Lot | 93.78 | 90.23 |
| 5 | On Painted Island / Chevron Lines | 95.23 | 94.36 |
| 6 | Out of Boundary | 80.36 | 77.11 |
| 7 | Legal | 96.63 | 92.26 |
| | | **Average Precision** ( % ) | 89.83 |
| | | **Average Recall** ( % ) | 91.03 |
| | | **F-Score** ( % ) | 90.42 |


Fig. 12: Detection performances along with relative precision and recall values obtained by different starting learning rates with SGD or Adam optimizer.

characteristics of the dataset and multiple trials. The goal of this subsection is to build the hyper-parameters setting, including the learning rate and the optimizer using our constructed dataset as an example. Fig. 12 presents the *F-Score* along with relative *Average Precision* and *Average Recall* values obtained by four combinations of optimizers with different starting learning rates. Among these four, the highest *F-Score* is obtained using SGD optimizer with starting learning rate $\eta = 0.001$. Results obtained from SGD optimizer with starting learning rate $\eta = 0.0001$ are not good. This phenomenon may result from the fact that too small learning rates will make the model trapped in the suboptimal solution and miss the global optimum. Moreover, as the training further continues, the model turns more susceptible to overfitting. Besides, using SGD as the optimizer with starting learning rate $\eta = 0.001$ generates more competitive performance in *F-Score* compared with Adam optimizer. According to Im $et\ al.$ [66] and Keskar $et\ al.$ [67], in the late stages of the training, step sizes learned by Adam optimizer may circumstantially become extremely small, making it suffering from inefficient convergence and being outperformed by SGD for final weights generation.

### *E. Evaluation of Illumination Robustness*

Since the proposed algorithm is expected to work all day in practical applications, its performance is evaluated with images

TABLE VII: Time complexity result comparisons

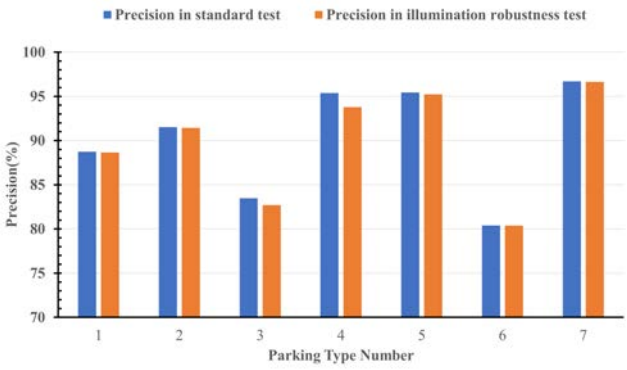
| Detection Model Name | Offline Training Time (hour) | Detection Time per Image (second) | FLOPs (G)(≈) |
|---|---|---|---|
| ResNet-50 [57] | 27.52 | 0.35 | 832.43 |
| ResNet-101 [57] | 62.68 | 0.40 | 907.21 |
| RetinaNet-50 [58] | 9.86 | 0.21 | 203.02 |
| RetinaNet-101 [58] | 21.24 | 0.24 | 279.38 |
| SSD [44] | 37.65 | 0.17 | **61.58** |
| **The Proposed Algorithm** | **6.23** | **0.07** | 65.64 |

of different illumination conditions. The brightness of tested images is randomly changed within a range to simulate different illumination conditions. TABLE VI shows the performance of the proposed algorithm in random illumination conditions. Fig. 13 presents the comparison of precision and recall results between the standard test and the illumination robustness test. Please take note that the experiment results of the standard test are presented in TABLE IV. The bars in blue represent results produced from the standard test, and bars in orange represent results produced from the illumination robustness test. Except for the brightness of test images, other factors of the test dataset remain the same.

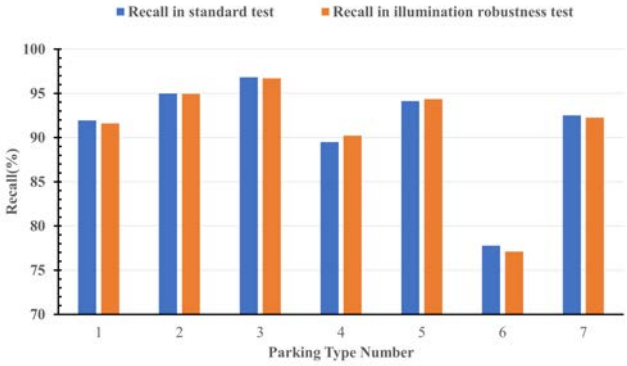
Observed from TABLE VI, the overall *F-Score* in various illumination conditions is 90.42%, very close to that of the standard test (i.e., 90.66% shown in TABLE IV). The *Average Precision* and *Average Recall* values of the seven parking types for illumination robustness test are 89.83% and 91.03%, respectively. These two values are about 1% less than those of the standard test. This phenomenon means that the changes of illumination will have some impacts on the performance of the detection of illegal parking types, which is expected. To be specific, deficient or excessive illumination condition will make small corner boundaries of parking lots and the blocked vehicles more blurry, which becomes similar to backgrounds. In summary, results of two kinds of tests are close, showing that the proposed algorithm exhibits satisfactory illumination robustness. Visualization examples of comparison results in the evaluation of illumination robustness are shown in Fig. 14.

### *F. Evaluation of Robustness under Different Weather Conditions*

Since the proposed detection algorithm with in-vehicle cameras is expected to work in urban environments regardless of weather conditions, its performance robustness needs to be

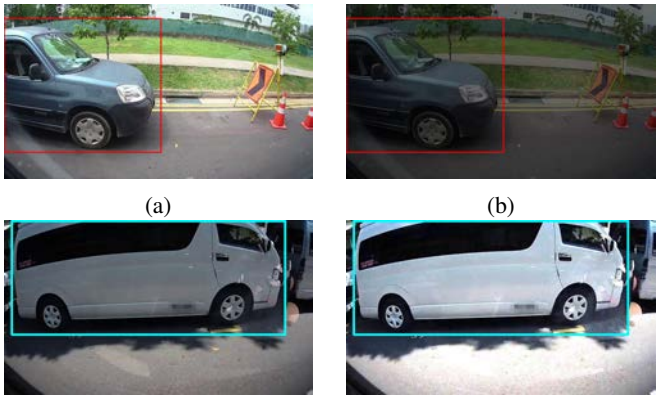



(a)




(b)

Fig. 13: Comparison of classification precision and recall results. (a) precision (b) recall.



(a) (b) (c) (d)

Fig. 14: Visualization of several comparison examples in the illumination robustness test. (a) and (c) are reference results in the standard test. (b) and (d) are corresponding results in the illumination robustness test. Red represents the $3^{rd}$ illegal parking type: "Along Double Yellow Lines" and Cyan represents the $6^{th}$ illegal parking type: "Out of Boundary".

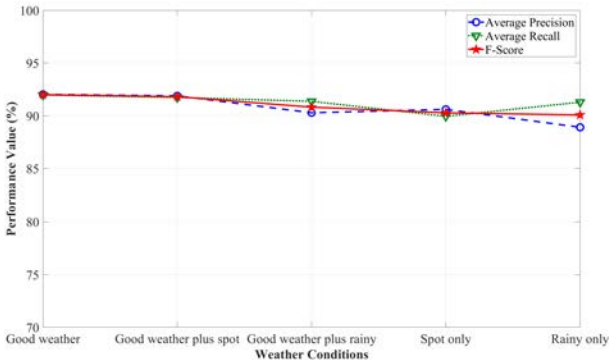



Fig. 15: Detection performances on average precision, average recall and F-score values under different weather conditions.

evaluated under various common weather conditions, such as good weather, rainy, or sunspot under extreme sunshiny days. The detection performances of the developed algorithm are expected to be affected, as raindrops, sunspots or reflections on the vehicle windows may blur images acquired by the in-vehicle cameras. We also add two other categories named as good weather plus rainy, and good weather plus spot, which combine images under good weather with rainy, and good weather with sunspots (or spot in short), respectively. It is observed from the performances in the five categories shown in Fig. 15, the performances of the proposed algorithm remain robust and stable in these common weather conditions. Specifically, the drop of the *F-Score* performance is less than $2\%$ comparing good weather to sunspots or rainy. It is further observed that the negative impact of rainy condition is more significant than that of the sunspots. This phenomenon is reasonable, because raindrops on the entire vehicle window are likely to affect the window visibility for the in-vehicle cameras, while the sunspots or reflections on vehicle windows may only affect parts of the windows. The evaluation results illustrate the detection of the proposed algorithm is robust under different weather conditions.

### G. *Limitation*

According to results shown in Fig. 13, if the illumination conditions are changing to too bright or too dark, the corner boundaries of parking lots will be blurrier, which becomes similar to backgrounds. It results in the decrease of the precision and recall values for the illegal parking type "Out of Boundary". As such, more image enhancement or super resolution techniques can be added in the overall real-time detection algorithm to better reduce the negative impacts generated by the excessive change of illumination conditions. In addition, in real-world environments, vehicles which are far away from the in-vehicle cameras are of very small shapes. Road lines or parking lots are also very hard to be recognized from far distance. As such, those vehicles parking far away from the in-vehicle cameras will be difficult to detect accurately by the proposed method. In the future, advanced image enhancement techniques could be added to improve the detection of vehicles parking far away.

## VI. CONCLUSIONS

To better deal with illegal parking detection tasks, a novel voting-based algorithm has been proposed to achieve real-time illegal parking detection. The new detection algorithm provides multiple illegal parking offences' types using in-vehicle cameras and is more suitable for real-world dynamic scenarios. It overcomes the limitation of existing CCTV based methods, which require complex and high-cost installation procedures for data collection but do not provide illegal parking details. A well-constructed illegal parking detection dataset has been newly developed by the research team, which has more than 10,000 images captured in Singapore with highly accurate annotations, covering six common illegal parking types and one legal parking type. In order to make the image labelling tasks less labour demanding, as well better relating the vehicle to its detailed parking type, a novel image labelling method named "minimal illegal units" is proposed in this paper. It marks multiple types of information in one rectangular box without separating their boundaries. Experiments have been conducted to evaluate the performance of the proposed solution on data collected in urban areas of Singapore. Benchmark results for illegal parking detection show that the proposed algorithm has a notable ability to detect illegal parking activities. It is able to further classify them into six types of illegally parking offences with high robustness with respect to illumination.

The limitation of the current illegal parking detection algorithm is all the videos were taken in Singapore, and mostly during daytime. In the future, more data from other countries and night time will be collected and evaluated by refining the detection algorithm presented in this paper. In addition, advanced image enhancement techniques can be added in the overall real-time detection algorithm to reduce negative impacts caused by the excessive changes of illumination conditions.

## ACKNOWLEDGMENT

The authors would like to thank the editors and reviewers for constructive comments which have helped improve the quality of the paper significantly. They would also like to express their thanks to the Urban Redevelopment Authority (URA) of Singapore, for their supports and help in this research.

## REFERENCES

[1] F. Shi, D. Wu, D. I. Arkhipov, Q. Liu, A. C. Regan, and J. A. McCann, "Parkcrowd: Reliable crowdsensing for aggregation and dissemination of parking space information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4032–4044, 2018.

[2] F. Bock, S. Di Martino, and A. Origlia, "Smart parking: Using a crowd of taxis to sense on-street parking space availability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 496–508, 2019.

[3] X. Ye, J. Wang, T. Wang, X. Yan, Q. Ye, and J. Chen, "Short-term prediction of available parking space based on machine learning approaches," *IEEE Access*, vol. 8, pp. 174 530–174 541, 2020.

[4] T. Fiez and L. J. Ratliff, "Gaussian mixture models for parking demand data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3571–3580, 2020.

[5] K. Gkolias and E. I. Vlahogianni, "Convolutional neural networks for on-street parking space detection in urban networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4318–4327, 2019.

[6] D. Zhao, C. Ju, G. Zhu, J. Ning, D. Luo, D. Zhang, and H. Ma, "Mepark: Using meters as sensors for citywide on-street parking availability prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[7] D. Lim and D. Park, "Ai analysis of illegal parking data at seocho city," in *Data Science and Digital Transformation in the Fourth Industrial Revolution*. Springer, 2021, pp. 165–178.

[8] R. Akhawaji, M. Sedky, and A.-H. Soliman, "Illegal parking detection using gaussian mixture model and kalman filter," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications*. IEEE, 2017, pp. 840–847.

[9] M. Sarker, K. Mostafa, C. Weihua, and M. K. Song, "Detection and recognition of illegally parked vehicles based on an adaptive gaussian mixture model and a seed fill algorithm," *Journal of information and communication convergence engineering*, vol. 13, no. 3, pp. 197–204, 2015.

[10] J. T. Lee, M. S. Ryoo, M. Riley, and J. Aggarwal, "Real-time illegal parking detection in outdoor environments using 1-d transformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1014–1024, 2009.

[11] C.-K. Ng, S.-N. Cheong, W.-J. Yap, and Y.-L. Foo, "Outdoor illegal parking detection system using convolutional neural network on raspberry pi," *International Journal of Engineering & Technology*, vol. 7, no. 3.7, pp. 17–20, 2018.

[12] X. Xie, C. Wang, S. Chen, G. Shi, and Z. Zhao, "Real-time illegal parking detection system based on deep learning," in *Proceedings of the 2017 International Conference on Deep Learning Technologies*. ACM, 2017, pp. 23–27.

[13] H. Tang, A. Peng, D. Zhang, T. Liu, and J. Ouyang, "Ssd real-time illegal parking detection based on contextual information transmission," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 62, no. 1, pp. 293–307, 2020.

[14] W. Wahyono and K. H. Jo, "Cumulative dual foreground differences for illegally parked vehicles detection," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017.

[15] P. Birch, W. Hassan, N. Bangalore, R. Young, and C. Chatwin, "Stationary traffic monitor," in *4th International Conference on Imaging for Crime Detection and Prevention 2011*. IET, 2011, pp. 1–6.

[16] D. Mochizuki, Y. Yano, T. Hashiyama, and S. Okuma, "Pedestrian detection with a vehicle camera using fast template matching based on background elimination and active search," *Electronics and Communications in Japan (Part II: Electronics)*, vol. 90, no. 10, pp. 115–126, 2007.

[17] J. Leng, Y. Liu, D. Du, T. Zhang, and P. Quan, "Robust obstacle detection and recognition for driver assistance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1560–1571, 2020.

[18] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara, "Rainy weather recognition from in-vehicle camera images for driver assistance," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* IEEE, 2005, pp. 205–210.

[19] N. Hautière, J.-P. Tarel, and D. Aubert, "Towards fog-free in-vehicle vision systems through contrast restoration," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[20] J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition," in *2000 IEEE Intelligent Transportation Systems. Proceedings*. IEEE, 2000, pp. 52–57.

[21] A. Ruta, F. Porikli, S. Watanabe, and Y. Li, "In-vehicle camera traffic sign detection and recognition," *Machine Vision and Applications*, vol. 22, no. 2, pp. 359–375, 2011.

[22] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[23] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3d localisation," *Machine vision and applications*, vol. 25, no. 3, pp. 633–647, 2014.

[24] A. Matsuda, T. Matsui, Y. Matsuda, H. Suwa, and K. Yasumoto, "A method for detecting street parking using dashboard camera videos," *Sensors and Materials*, vol. 33, no. 1, pp. 17–34, 2021.

[25] A. H. Sodhro, S. Pirbhulal, and V. H. C. De Albuquerque, "Artificial intelligence-driven mechanism for edge computing-based industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4235–4243, 2019.

[26] T. Zhang, A. H. Sodhro, Z. Luo, N. Zahid, M. W. Nawaz, S. Pirbhulal, and M. Muzammal, "A joint deep learning and internet of medical things driven framework for elderly patients," *IEEE Access*, vol. 8, pp. 75 822–75 832, 2020.

[27] A. H. Sodhro, G. H. Sodhro, M. Guizani, S. Pirbhulal, and A. Boukerche, "Ai-enabled reliable channel modeling architecture for fog computing

vehicular networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 14–21, 2020.

[28] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[29] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *arXiv preprint arXiv:1704.05519*, 2017.

[30] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 4, pp. 349–361, 2001.

[31] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, "Shape-based pedestrian detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*. IEEE, 2000, pp. 215–220.

[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[33] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 12, pp. 2179–2195, 2008.

[34] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[35] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[36] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[37] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[38] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[39] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning*, 2010, pp. 807–814.

[43] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[46] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Mini-yolov3: real-time object detector for embedded applications," *IEEE Access*, vol. 7, pp. 133 529–133 538, 2019.

[47] J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng, "Fabric defect detection using the improved yolov3 model," *Journal of Engineered Fibers and Fabrics*, vol. 15, p. 1558925020908268, 2020.

[48] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, "Yolo-tomato: A robust algorithm for tomato detection based on yolov3," *Sensors*, vol. 20, no. 7, p. 2145, 2020.

[49] A. Kuznetsova, T. Maleva, and V. Soloviev, "Using yolov3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot," *Agronomy*, vol. 10, no. 7, p. 1016, 2020.

[50] D. K. Prasad, "Survey of the problem of object detection in real images," *International Journal of Image Processing (IJIP)*, vol. 6, no. 6, p. 441, 2012.

[51] M. S. Al-Shemarry, Y. Li, and S. Abdulla, "An efficient texture descriptor for the detection of license plates from vehicle images in difficult conditions," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[52] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[53] G. Montavon, G. Orr, and K.-R. Müller, *Neural networks: tricks of the trade*. springer, 2012, vol. 7700.

[54] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3908–3916.

[55] H. Ma, Y. Liu, Y. Ren, and J. Yu, "Detection of collapsed buildings in post-earthquake remote sensing images based on the improved yolov3," *Remote Sensing*, vol. 12, no. 1, p. 44, 2020.

[56] J. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, no. 1, pp. 66–77, 2012.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[58] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[60] Y. Han, X. Liu, Z. Sheng, Y. Ren, X. Han, J. You, R. Liu, and Z. Luo, "Wasserstein loss-based deep object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 998–999.

[61] Q.-J. Wang, S.-Y. Zhang, S.-F. Dong, G.-C. Zhang, J. Yang, R. Li, and H.-Q. Wang, "Pest24: A large-scale very small object data set of agricultural pests for multi-target detection," *Computers and Electronics in Agriculture*, vol. 175, p. 105585, 2020.

[62] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European Conference on Information Retrieval*. Springer, 2005, pp. 345–359.

[63] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[64] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[66] D. J. Im, M. Tao, and K. Branson, "An empirical analysis of the optimization of deep network loss surfaces," *arXiv preprint arXiv:1612.04010*, 2016.

[67] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to sgd," *arXiv preprint arXiv:1712.07628*, 2017.

**Xinggan Peng** received the B.Eng. degree (Hons.) from the University of Electronic Science and Technology of China and the University of Glasgow in 2017 and the M.S. degree in electrical and computer engineering from The Ohio State University in 2019. He is currently pursuing the Ph.D. degree with the School of EEE, Nanyang Technological University. His research interests include intelligent transportation systems and deep learning.

**Rongzihan Song** (Graduate Student Member, IEEE) received the B.Eng. degree from the Changsha University of Science and Technology in 2018 and the M.S. degree from Nanyang Technological University in 2019, where he is currently pursuing the Ph.D. degree with the School of EEE. His research interests include AIoT, machine vision, and deep learning.

**Qi Cao** received the Bachelor of Engineering degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2000, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently an Assistant Professor with the School of Computing Science, University of Glasgow, Singapore Campus. His research interests include virtual reality, image processing, data analytics, and computational intelligence.

**Yue Li** received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2015 and 2020, respectively. He is currently the Chief AI Scientist of I Innovations Private Ltd., Singapore. His research interests include AIoT and predictive maintenance in smart industry.

**Dongshun Cui** received the B.Eng. degree from the South China University of Technology in 2012, the M.S. degree from the Beijing Institute of Technology in 2014, and the Ph.D. degree from Nanyang Technological University, Singapore. He is currently the CTO of Mind PointEye. His research interests include machine learning and computer vision.

**Xiaofan Jia** (Graduate Student Member, IEEE) received the B.Eng. degree from the Nanjing University of Science and Technology, China, in 2014, and the M.Sc. degree from Nanyang Technological University, Singapore, in 2018, where she is currently pursuing the Ph.D. degree with the School of EEE. She is a Research Associate at the School of EEE, Nanyang Technological University. Her current research interests include machine learning, computer vision, and computational electromagnetics.

**Zhiping Lin** (Senior Member, IEEE) received the Ph.D. degree in information engineering from the University of Cambridge, U.K., in 1987. Since 1999, he has been an Associate Professor at the School of EEE, Nanyang Technological University, Singapore. Prior to that, he has worked at the DSO National Laboratories, Singapore; and Shantou University, China. He has published about 200 journal articles and over 200 conference papers. His research interests include signal processing and machine learning. He was the Editor-in-Chief of Multidimensional Systems and Signal Processing for the period of 2011–2015. He has served as an associate editor for several other international journals. He was a Distinguished Lecturer of the IEEE Circuits and Systems Society for the period of 2007–2008.

**Guang-Bin Huang** (Senior Member, IEEE) was a Full Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently the Founder of Mind PointEye, Singapore. He has been listed in Thomson Reuters's "The World's Most Influential Scientific Minds" in the past several years since 2014. His two works on extreme learning machines (ELM) have been listed by Google Scholar in 2017 as top two and top seven, respectively, in its "Classic Papers: Articles That Have Stood The Test of Time," as top ten in artificial intelligence. He was a Nominee of the Singapore President Science Award (2016, 2017, 2018, and 2019). He was awarded by Thomson Reuters as the "Highly Cited Researcher" (in two fields: engineering and computer science). He has received the Best Paper Award from IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (2013).