A New CNN-Based Method for Multi-Directional Car License Plate Detection

Lele Xie, Tasweer Ahmad, Lianwen Jin, *Member, IEEE*, Yuliang Liu, and Sheng Zhang

***Abstract*— This paper presents a novel convolutional neural network (CNN) -based method for high-accuracy real-time car license plate detection. Many contemporary methods for car license plate detection are reasonably effective under the specific conditions or strong assumptions only. However, they exhibit poor performance when the assessed car license plate images have a degree of rotation, as a result of manual capture by traffic police or deviation of the camera. Therefore, we propose the a CNN-based MD-YOLO framework for multi-directional car license plate detection. Using accurate rotation angle prediction and a fast intersection-over-union evaluation strategy, our proposed method can elegantly manage rotational problems in real-time scenarios. A series of experiments have been carried out to establish that the proposed method outperforms over other existing state-of-the-art methods in terms of better accuracy and lower computational cost.**

***Index Terms*— License plate detection, convolutional neural network, MD-YOLO, intersection over union, multi-direction.**

I. Introduction

IN RECENT years, the number of privately-owned cars has increased considerably and this has, in turn, exacerbated the traffic management burden. The resultant congestion has caused extreme problems, such as traffic accidents or public-space vulnerability to crime or terrorist attacks. Physical management of this ubiquity of automobiles is quite challenging, and this problem has encouraged the development of automatic systems to manage traffic jams. In particular, automatic license plate detection can adequately monitor the cars and greatly alleviate traffic management burdens; thus, this approach has attracted considerable attention of researchers [14], [23], [31]. Further, automatic car license plate detection can also be applied in many other scenarios, e.g., for expressway toll collection, overspeeding violations [20], surveillance and management of unattended parking lots.

Both traditional and convolutional neural network (CNN)-based methods are being used to solve the problem of car license plate detection. The traditional methods involve hand-crafted features such as color- [1], edge- [40] and morphology- [16], which are primarily confined by stringent conditions. For example, some of these systems need high resolution images as input, the acquisition of which requires expensive equipment, while others require strict device mounting free from translation and rotation. However, real-world scenarios are quite different where car license plate detection becomes very challenging [39]. Different kinds of cars and roads, changing weather conditions, camera-device rotation, and so on curtail the detection performance significantly. Thus, under the complex situation, it is relatively difficult to propose a robust method with hand-crafted features. Although people can employ multiple independent features and incorporate some models together, as mentioned in [34], it is still hard to distinguish whether it is enough to meet the challenge with such limited features and models. To alleviate these problems, CNN-based methods [18] have been devised, which automatically learn features from the acquired data. These kinds of detection methods have recently yielded very impressive results [4], [15], [19], [28]; however, their time consumption is significantly higher than those of the aforementioned techniques. Further, despite the viability of both traditional and CNN-based methods, the problems associated with multi-directional (MD) car license plate detection have not yet been satisfactorily resolved to date, because of the difficulties due to the viewpoint variation of hand-held cameras or the accidental rotation of mounted cameras.

Inspired by the "you only look once" (YOLO) [27] framework, we propose a CNN-based method that can manage the multi-directional problem reasonably well. We refer to this method as "MD-YOLO". The main contributions of our work are summarized as follows:

1) We propose a novel accurate rotation angle prediction method to realize multi-directional car license plate detection;
2) To rapidly evaluate the intersection-over-union (IoU) between two rotational rectangles, we propose an approximate method, namely, the angle deviation penalty factor (ADPF);
3) To further promote the detection accuracy, we design a prepositive CNN model that is implemented before MD-YOLO, which serves to determine the "attention region" in the overall framework. The method of cascading the two models is based on *prior knowledge*: as the car license plates are fixed on the cars, some dis-

Manuscript received April 30, 2017; revised October 8, 2017 and November 20, 2017; accepted December 2, 2017. Date of publication January 11, 2018; date of current version February 1, 2018. This work was supported in part by the National Key Research and Development Plan of China under Grant 2016YFB1001405, in part by the GD-NSF under Grant 2017A030312006, in part by the GDSTP under Grant 2015B010131004 and Grant 2015B010101004, and in part by the GZSTP under Grant 201607010227. The Associate Editor for this paper was Q. Wang. *(Corresponding author: Lianwen Jin.)*

The authors are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: xie.lele@mail.scut.edu.cn; tasveerahmad@gmail.com; eelwjin@scut.edu.cn; liu.yuliang@mail.scut.edu.cn; zsscut@sina.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2784093

tance will inevitably exist between any two plates. The synergy of this concept is explained in the subsequent section;

4) The proposed method achieves state-of-the-art detection accuracy and can also be run in real time.

The remainder of this paper is organized as follows. Section II gives a brief review of car license plate detection. In Section III, the proposed method is described in detail. Section IV presents a series of experimental results and analyses. Finally, a concise conclusion summarizing the main points of the entire work is supplied in Section V.

II. Related Work

Over the last two decades, impressive research work has been carried out by the computer vision community to address the problem of automatic car license plate detection and recognition [8]. This task of license plate detection can broadly be categorized into following three approaches; 1) Region-based approaches, 2) Pixel-to-Pixel approaches [22], and 3) Color-based approaches [41]. In region-based methods, input image is segmented into smaller regions, where some pre-specified attributes of the license plate are located in such subsequent regions. Numerous unified approaches, based on morphological and high-pass filtering, are devised to effectively perform car license plate detection task in [11], [21], [25], [37], and [44]. In pixel-to-pixel approach, every pixel is evaluated in the image among its neighboring pixels to form a coarse rectangular box. The whole input image is scanned pixel-by-pixel using a detection window. The response of scanning window is computed at each pixel location and the regions with high response to scanning window are selected as candidate region. In this regard, L. Dlagnekov (2004) formulated an Adaboost classifier for pixel-to-pixel classification of an image [6]. In [10], Ho *et al.* addressed the problem of license plate detection using pixel-to-pixel method. The authors contributed by devising the Adaboost classifier at first step and then SIFT-based SVM classifier in the later stage. Scale-invariant Feature Transform (SIFT) based approach was also exercised by F.A. Silva for license plate detection and recognition [3]. In [36], HOG-features of the segmented license plate were extracted and then a Gentle AdaBoost trained classifier was used for license plate detection. Subsequently, a novel color-based approach was developed by Azad *et al.* [2], by converting the RGB-color images to the HSV-color space. Then, this HSV-image is segmented into smaller blocks, where each block is examined for license plate or some part of license plate by using carefully formulated filtering procedure. Deb *et al.* in [5] proposed three step procedures for car license plate detection and recognition; where first, i) sliding concentric windows (SCWs) based method is used for detection of candidate regions, ii) HSI-color model based verification is rendered to identify candidate regions, and finally iii) candidate regions are decomposed using position histogram to segment out alphanumeric characters in the plate.

Over the couple of last few years, many computer vision problems are being addressed by using deep convolutional neural networks (CNNs); in line with that, many researchers exercised to solve the problems of car license plate detection and recognition using CNNs [18], [26]. CNNs have already exhibited remarkable performance for text and optical character recognition [13], [30], [38] and object detection [14], [19], [27], [28]; whereas, the ancillary task of license plate detection and localization is underway to be better addressed by using deep CNNs; especially in the challenging situation of multi-directional car plate detection, current state-of-the-art CNN based models may not be directly used to achieve good enough performance. Although the problem of multi-directional car license plate detection was aware and has been addressed by some researchers (eg. [12]), but the performance using traditional method [12] is still far away from practical promising.

Recently, researches of computer vision tasks in transportation system have made significant progresses. Under high demand of robustness, some new methods [32], [33] tend to employ the features extracted by CNN instead of hand-crafted features. Besides, it is necessary to enrich the dataset with certain data augmentation strategies. The work of [33], balancing the proportion of different objects in dataset with hierarchical augmentation, has obviously improved the accuracy. Furthermore, many researchers have tried utilizing the prior knowledge e.g. location [32], symmetry [42] to improve the performance. Our proposed method in this paper also follows these trends to achieve high detection accuracy.

III. Methodology

In this section, we introduce our MD-YOLO method and discuss a further refinement of the proposed design.

A. Multi-Directional Car Plate Detection With MD-YOLO

YOLO [27] is a unified detection framework based on CNN, which regards detection as a regression problem. With some considerable improvements in the original framework, our proposed MD-YOLO can treat multi-directional problems with high detection accuracy. The MD-YOLO network structure is shown in Fig. 1.

1) Rotation Angle: The original YOLO framework predicts the center coordinate, height, and width of each object only. However, the method proposed in this paper introduces angle information and guides the model to regress and determine the angle of rotation of a given car license plate image. Thus, before training the model, we should parameterize the original angle information. The tangent value is adopted to parameterize the angle θ, using the formula given by (1).

$$a_i = \frac{\tan\theta_i}{\max\limits_{1\le k\le N}\{\mathrm{abs}(\tan\theta_k)\}} \tag{1}$$

where i indicates the i-th car plate. N is the total number of car plates in training set, and abs is the absolute value function. We enumerated each rotation angle θ_k of car license plate in training set to find maximum rotation angle. Note that the maximum rotation angle in training set should be much bigger than the one in test set so as to accommodate to the actual

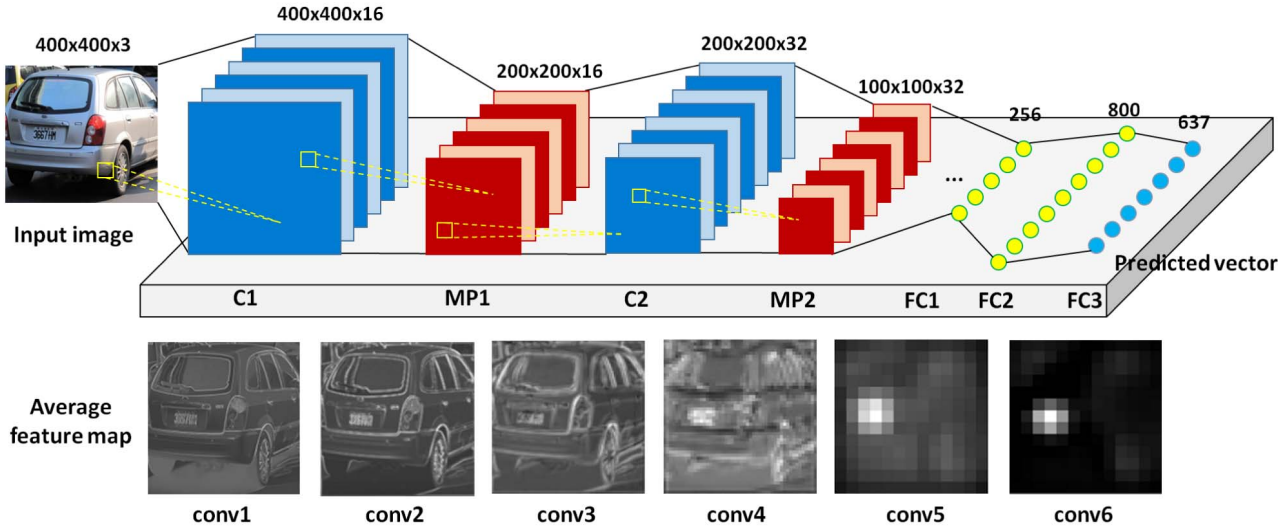

Fig. 1. The model of MD-YOLO. The average feature maps of each convolutional layer are presented here. From the visualization, our CNN model typically has strong response in the position of car license plate.

situation, which can be guaranteed through random rotation strategy as mentioned later. Different from the other parameters considered in this method, we choose to retain the negative value for the angle variable, which allows determination of the direction of a slanted car plate. Hence, all angle values are normalized within the interval $[-1, +1]$.

2) Evaluating IoU With ADPF: We wish to predict the rotation angle, which involves evaluation of the IoU between two rotational rectangles during forward propagation of the CNN. We attempt to evaluate the IoU using the Computational Geometry Algorithms Library (CGAL) [24] directly. However, this method is computationally extensive and has poor efficacy for real-time detection systems, despite its higher precision. Thus, we propose a simple and efficient method of approximately computing the IoU using the angle deviation penalty factor (ADPF), which is a decreasing function of the angle deviation. That is, a larger angle deviation yields smaller ADPF. The definition of ADPF is given by

$$P_{ij} = 1 - \text{abs}\left(\frac{\theta_i - \theta_j}{\max .angle}\right) \tag{2}$$

$$\hat{IoU}_{ij} = P_{ij} * IoU^*_{ij} \tag{3}$$

where θ_i and θ_j are the rotation angles of the i-th and j-th rotational rectangles. During the forward propagation, we need to compute the IoU between predicted bounding box and ground truth, so one of these two rotation angles comes from the predicted values in corresponding predicted bounding box and the other one comes from the angle labels in training set. P_{ij} is ADPF between rotational rectangles i and j. The term max.$angle$ is a constant factor that guarantees a positive value for the ADPF which is set to be $\pi/1.5$ in our experiment. To compute IoU^*_{ij}, the two rotational rectangles are first made horizontal by neglecting their angles. Then, their overlap is computed. Finally, we compute the approximate value $\hat{IoU}_{ij}$ using the ADPF, as shown in eq. (3). Table I presents a latency comparison between calculations using the ADPF and CGAL; hence, it is apparent that the ADPF case exhibits superior performance.

TABLE I
CGAL VS. ADPF TIME CONSUMPTION FOR IOU CALCULATION. FOR EACH ITERATION, THE IOU IS CALCULATED BETWEEN A PAIR OF ROTATIONAL RECTANGLES

Method	10^3 iterations	10^6 iterations
CGAL	5.15s	>1h
ADPF (our)	<5us	2.6ms

3) Regression and Detection: In the regression process, we wish to find a non-linear function $F(\boldsymbol{I})$ that accepts image $\boldsymbol{I}$ as input and produces a specific target $\boldsymbol{t}$. However, it is extremely difficult to determine $F(\boldsymbol{I})$ exactly; therefore, we employ CNN to allow the algorithm to learn an approximate function $\hat{F}(\boldsymbol{I}, \boldsymbol{\Omega})$. Further, we use the back propagation algorithm (BP) to approach $F(\boldsymbol{I})$. These variables and methods are described mathematically as follows:

$$\boldsymbol{t} = F(\boldsymbol{I}),\ \boldsymbol{v} = \hat{F}(\boldsymbol{I}, \boldsymbol{\Omega}) \tag{4}$$

$$loss: J = J(\boldsymbol{t} - \boldsymbol{v}) \tag{5}$$

$$\boldsymbol{\Omega}^{i+1} = \boldsymbol{\Omega}^i - \alpha \frac{\partial J}{\partial \boldsymbol{\Omega}}, (\alpha > 0) \tag{6}$$

$$\hat{F}(\boldsymbol{I}, \boldsymbol{\Omega}^*) \overset{approach}{\longrightarrow} F(\boldsymbol{I}) \tag{7}$$

where $\boldsymbol{v}$ is the predicted CNN value and i represents the i-th iteration of BP. $\boldsymbol{\Omega}^*$ denotes the final optimized CNN weight parameter and J is the loss function used for the BP algorithm.

MD-YOLO executes direct regression to detect a car license plate. This regression is schematically illustrated in Fig. 2. Each input image is divided into regular $S \times S$ grid cells, and the cell in which the car plate center is located is used to detect the car license plate. B bounding boxes and a confidence score $P(object)$ are predicted for each grid cell. The confi-

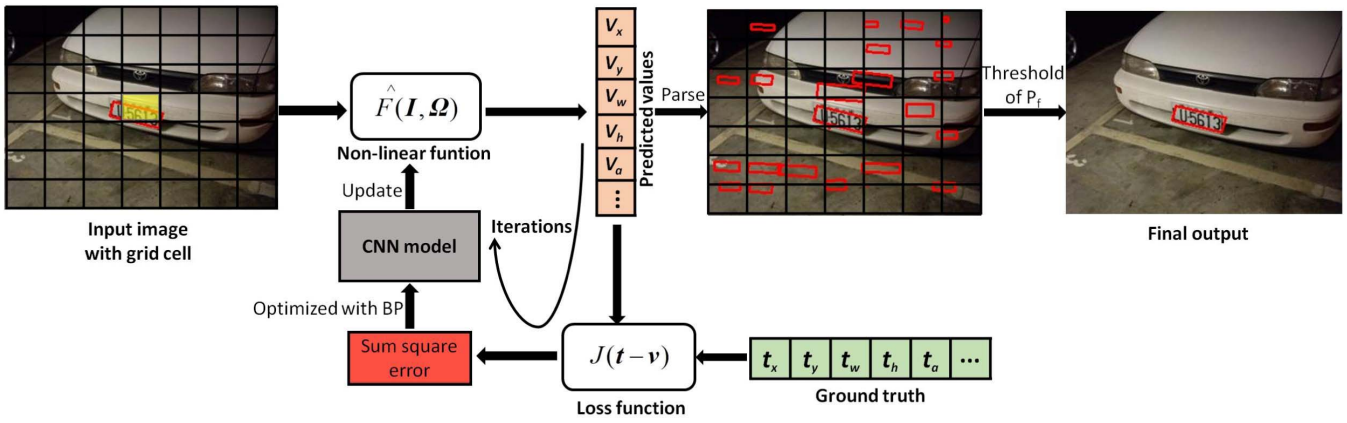

Fig. 2. MD-YOLO regression procedure. Each grid cell predicts two bounding boxes, some of which are neglected for low confidence. As the geometric center of the plate is located in the yellow grid cell, that cell predicts the true car license plate bounding box. The predicted box with P_f larger than the threshold is selected to be the final output.

dence reflects how likely the bounding boxes will contain car license plate. Further, each bounding box predicts six values: x, y, *width*, *height*, *angle* and *confidence* $P(IoU)$. Here, x and y are coordinates and *confidence* $P(IoU)$ represents the IoU between the bounding box and ground truth. Thus, $P(object) \times P(IoU)$ yields the final output probability P_f of each bounding box. If P_f is larger than a certain threshold, the corresponding bounding box is selected as the output. The regression targets $\boldsymbol{t} = \{t_x, t_y, t_w, t_h, t_a\}$ and predicted values $\boldsymbol{v} = \{v_x, v_y, v_w, v_h, v_a\}$ are parameterized as follows:

$$t_x = \frac{(x^* - x_g)S}{I_w}, \quad t_y = \frac{(y^* - y_g)S}{I_h}, \quad t_w = \sqrt{\frac{w^*}{I_w}}, \quad t_h = \sqrt{\frac{h^*}{I_h}} \tag{8}$$

$$v_x = \frac{(x - x_g)S}{I_w}, \quad v_y = \frac{(y - y_g)S}{I_h}, \quad v_w = \sqrt{\frac{w}{I_w}}, \quad v_h = \sqrt{\frac{h}{I_h}} \tag{9}$$

Here, x, y, w, h, and a represent the bounding box center coordinates, width, height, and rotation angle, respectively. The tuples $\{x, y, w, h, a\}$ and $\{x^*, y^*, w^*, h^*, a^*\}$ apply to the selected bounding box and ground truth, respectively. The point (x_g, y_g) has the coordinates of the top-left vertex of the grid cell corresponding to that box. The box rotation angle is parameterized as shown in (1).

When an image is passed through the CNN, the network extracts global features based on the entire image. Then, the final fully connected layer outputs a vector $\boldsymbol{d}$ with length 637, which encodes the predicted values. Note that the length can be calculated from $S \times S \times (B \times 6 + 1) = 7 \times 7 \times (2 \times 6 + 1) = 637$. The predicted vector is parsed in Fig. 3. Therefore, for grid cell (i, j) in the image, the k-th bounding box $\{x_{ij}^k, y_{ij}^k, w_{ij}^k, h_{ij}^k, a_{ij}^k\}$ of this cell can be decoded as:

$$\boldsymbol{u}^l = \sigma_{leaky}(\mathbf{W_c}^l \boldsymbol{x}^{l-1} + \boldsymbol{b}^l) \tag{10}$$

$$\boldsymbol{d} = \sigma_{ident}(\mathbf{W_f}^{l+1} \boldsymbol{u}^l + \boldsymbol{b}^{l+1}) \tag{11}$$

$$ind = (i \times S + j) \times (B \times 6 + 1) \tag{12}$$

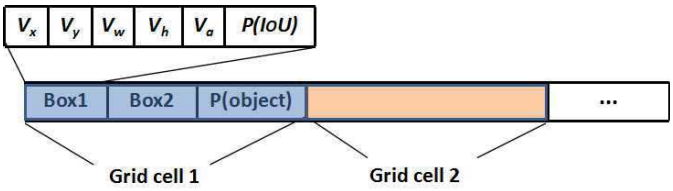

Fig. 3. Analysis of predicted vector. The predicted values of each grid cell are stored in sequence. There are two predicted bounding boxes and one confidence score $P(object)$ in a grid cell.

$$x_{ij}^k = \frac{d_{ind+6k} \times I_w}{S} + x_g \tag{13}$$

$$y_{ij}^k = \frac{d_{ind+6k+1} \times I_h}{S} + y_g \tag{14}$$

$$w_{ij}^k = d_{ind+6k+2}^2 \times I_w \tag{15}$$

$$h_{ij}^k = d_{ind+6k+3}^2 \times I_h \tag{16}$$

$$a_{ij}^k = \arctan\left(d_{ind+6k+4} \times \max_{1 \le k \le N}(\text{abs}(\tan\theta_k))\right) \tag{17}$$

Here, we symbolically demonstrate the MD-YOLO computational procedure using two layers, i.e., the $(l-1)$-th and l-th layers. $\mathbf{W_c}$ and $\mathbf{W_f}$ denote the weight parameters of convolutional and fully connected layers, respectively, whereas $\boldsymbol{x}$ and $\boldsymbol{u}$ represent the inputs from $(l-1)$-th and l-th layers, respectively. Further, $\mathbf{b}$ is the bias parameter, and the ind is the index of the predicted value in vector $\boldsymbol{d}$.

4) Network Design and Training: As shown in Fig. 1, our MD-YOLO network consists of 7 convolutional layers and 3 fully connected layers. All the convolutional layers employ a kernel size of 3×3 and a padding size of 1. The first five convolutional layers are followed by Max-Pooling (MP) with a 2×2 window size and a stride of 2. The final three fully connected (FC) layers have 256, 800, and 637 channels, respectively. To ensure that our CNN model can identify negative rotation angle values, leaky and identity functions are chosen as the activation functions, rather than ReLU function. We first pre-train our model using the ImageNet dataset [29]. Then, the model is trained to detect car license plates. We select the sum-squared error as the loss

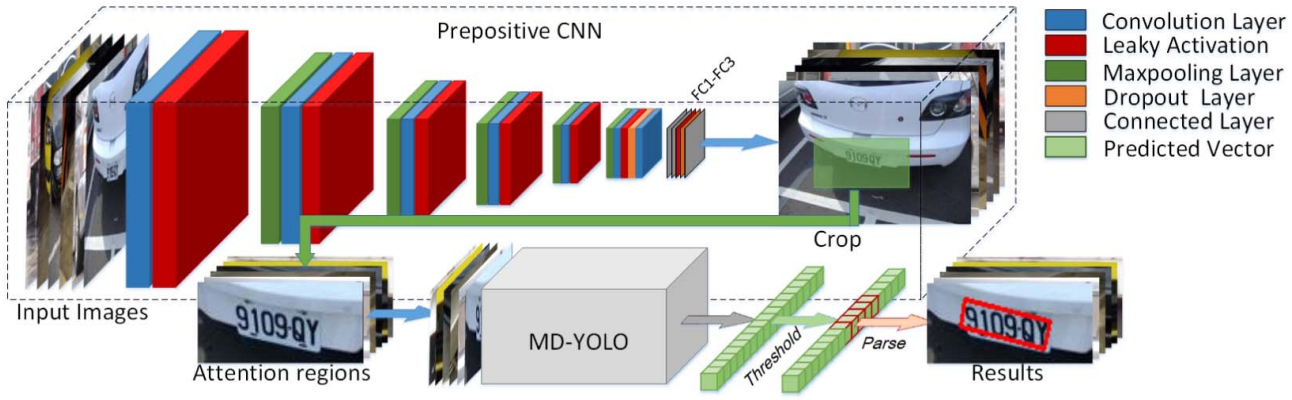

Fig. 4. Overall ALMD-YOLO framework. First, the prepositive CNN model takes a full image as input and produces an attention region. Then, the attention region is cropped out and input to MD-YOLO which finally determines a precise rotational rectangular region.

function; this is similar to the approach used in [27]. However, different from [27], we introduce the loss of the rotation angle in order to guide the model to facilitate angle determination. The loss function we used is defined as:

$$\begin{aligned} J = {} & \lambda_{coord} \sum_{i\in\{x,y,w,h,a\}} \sum_{j=0}^{S^2} \sum_{k=0}^{B} \delta_{jk}^{obj} (v_{ij} - t_{ij})^2 \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \delta_{ij}^{obj} (P_i(IoU) - IoU_{ij})^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \delta_{ij}^{noobj} P_i^2(IoU) \\ & + \sum_{i=0}^{S^2} \delta_i^{obj} (P_i(object) - 1)^2 \end{aligned} \tag{18}$$

where λ_{coord} and λ_{noobj} are 5 and 0.5, respectively. Further, if the geometric center of the plate is located in the i-th cell, $\delta_i^{obj} = 1$. IoU_{ij} is evaluated by considering the difference between the predicted bounding box and ground truth. Therefore, if the predicted value v deviates from target t, the square function $(v - t)^2$ will produce a loss that constrains the prediction through backward propagation. Further, if $\delta_i^{obj} = 1$ is satisfied, $P_i(IoU)$ and $P_i(object)$ are forced to become IoU and 1, respectively.

B. Further Refinement of Proposed Design

For car license plate detection using an acquired image, the image area proportion of the car license plate is usually very small. When the MD-YOLO method is applied, global image features are extracted, and small image portions such as the car license plate may deem to introduce some redundant information. However, this problem is alleviated by introducing a prepositive CNN attention model, which is employed prior to the implementation of MD-YOLO. Such attention models can remove redundant information and can surpass the performance significantly. We refer to the modified approach by employing the attention-like method within the overall framework as "ALMD-YOLO"; this framework is further discussed in the followings.

TABLE II
DETAILS OF REDESIGNED CNN MODEL FOR MD-YOLO

Layer Type	Parameters
Fully connected	#neurons:637
Dropout	Prop:0.3
Fully connected+Leaky	#neurons:985
Fully connected+Linear	#neurons:384
Convolution+Leaky	#filters:256, k: 3×3, p:1
Dropout	Prop:0.2
Convolution+Leaky	#filters:256, k: 3×3, p:1
Convolution+Leaky	#filters:164, k: 3×3, p:1
Convolution+Leaky	#filters:128, k: 3×3, p:1
Max pooling	k: 2×2, s:2
Convolution+Leaky	#filters:64, k: 3×3, p:1
Max pooling	k: 2×2, s:2
Convolution+Leaky	#filters:32, k: 3×3, p:1
Max pooling	k: 2×2, s:2
Convolution+Leaky	#filters:16, k: 3×3, p:1
Input	100×100

1) Role of Prepositive CNN: The prepositive CNN plays an important role in removing redundant information. When an image is taken as input, the prepositive CNN is expected to identify an attention region containing a car license plate, which is significantly smaller than the original image, having size $(2 \times width) \times (3 \times height)$ relative to the car plate. Then, the attention region is passed to MD-YOLO in order to determine a precise rotational rectangular region. The ALMD-YOLO framework is illustrated in Fig. 4.

2) Network Design: The prepositive CNN uses the same network as MD-YOLO (Subsection A), but the final fully connected layer is 539 pixels long. This is because the prepositive CNN does not predict the angle parameter. Further, we redesign a new CNN model with small input 100×100 for MD-YOLO. The details of the network are listed in Table. II.

3) Necessary Prior Knowledge: We consider a challenging scenario. That is, if multiple car license plates are contained within the attention region and some plates are cropped by the border, there may be some confusion as to whether the incomplete plates near the border should be detected. Further, if those partial plates are detected, there is no effective means of obtaining the complete plates. In such a scenario, attention-region-based detection becomes a complex and contradictory problem. Therefore, we employ prior knowledge: as the car license plates are fixed on cars, some distance will inevitably exist between any two plates. This knowledge can guarantee that attention regions contain one full car license plate only, which greatly simplifies the detection problem. Yuan *et al.* [42] have also shown that utilizing the prior knowledge about traffic signs can help improve the performance of computer vision tasks in transportation system.

4) ALMD-YOLO Limitation and Advantage: ALMD-YOLO has a drawback that it is a combination of two models (the prepositive model and MD-YOLO). These models should be trained separately, with independent loss functions and different datasets. Although such two-stage procedure is a little cumbersome, it actually is superior to the end-to-end model on detection accuracy. To prove this point, the next section presents an experiment of the end-to-end model.

IV. Experiments

A. Dataset

The Application Oriented License Plate (AOLP) [12] dataset which contains 2049 images of Taiwanese car plates, was employed in the car license plate detection experiment. This dataset is categorized into three subsets: access control (AC): 681 samples; traffic law enforcement (LE): 757 samples; and road patrol (RP): 611 samples. AC refers to cases in which a car traverses a fixed passage with a significantly lower speed than normal or comes to a full stop. LE refers to cases in which a car violates traffic laws, with this behavior being captured by a roadside camera. In this scenario, the background may be heavily cluttered, with road signs, pedestrians, or multiple plates in a single image. Finally, RP refers to cases in which the camera is fixed on a patrol car and the images are acquired from arbitrary viewpoints and distances. A detailed description of the AOLP dataset can be found in [12].

Because of the particular ground truth of AOLP [12], this dataset does not include an angle label, which means the label format of this dataset is $\{x, y, w_1, h_1\}$. Here, (x, y) is center coordinate and w_1, h_1 are width and height of car license plate. We call the original dataset {AC, LE, RP} as horizontal dataset. In fact, the RP subset of the AOLP dataset has many rotational car license plates; therefore, we re-labeled the RP subset (611 samples) with rotation angle information and the label format has changed to $\{x, y, w_2, h_2, a\}$, where a means the rotation angle. We use symbol "RP+" to represent the re-labeled RP subset. Finally, we call the new dataset {AC, LE, RP+} as rotational dataset. It is worth noticing that w_2, h_2 are different from w_1, h_1 because the re-labeled ground truth has become rotational rectangle, exactly enclosed by the original ground truth (the horizontal circumscribed rectangle).

TABLE III

Comparison Between Horizontal and Rotational Datasets. The Values are Expressed in REC/PREC/F-Measure Format

Data \ IoU	0.5	0.6	0.7
Horizontal	96.6/97.3/97.0	89.0/89.7/89.3	72.0/72.6/72.3
Rotational	92.6/94.4/93.5	80.5/82.1/81.3	60.0/61.2/60.6

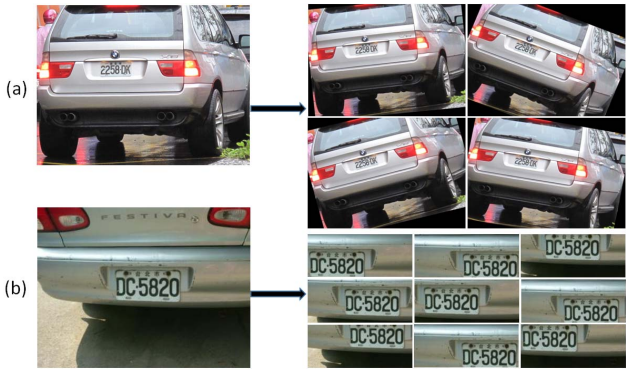

Fig. 5. (a) Random rotation strategy. Each image is randomly rotated within $[-30°, +30°]$. (b) Attention region training set for MD-YOLO. Each attention region has the size of $(2 \times width) \times (3 \times height)$ relative to the ground truth (car license plate).

After we re-labeled the RP subset with extra rotation angle information, our ground truth can enclose the car license plate tightly, without redundant region. As mentioned in [35], more information provided by dataset can generally improve the results of computer vision tasks. As there is no standard split for AOLP dataset, in this study, we randomly divided the data into training and test sets with a 2:1 ratio. When splitting the rotational dataset, we segmented the horizontal and rotational license plates in training set with ratio 2:1 too. Both of the horizonal and rotational datasets employ the same splitting method for fair comparison.

We implemented our MD-YOLO model on both the rotational dataset and the original horizontal dataset to conduct a comparison. The results are listed in Table III and we can see that detection is easier for the horizontal than the rotational dataset. However, our method can still achieve high detection accuracy when applied to the rotational dataset. Besides, all the other experiments were conducted on the rotational dataset.

Note that a random rotation data augmentation strategy can be adopted during the training process of MD-YOLO model, which improved the performance significantly. That is, we randomly rotated the images along the geometric center of the car plate in the AC and LE subset, within $[-30°, +30°]$. This process forced the pseudo samples to be similar to the RP subset, which had the largest rotation angle of 30°. The data augmentation strategy is illustrated in Fig. 5. Actually, the variation of rotation angles seen in training set is very important to the model robustness. If we have sufficient rotated-angled samples in training set, then the proposed model can fairly deal with real-world scenes. Therefore, we adopt such random rotation strategy, forcing the rotation angles to uniformly distribute in interval $[-30°, +30°]$.

TABLE IV

COMPARISON OF DETECTION ACCURACIES OF VARIOUS METHODS. ALL THE METHODS ARE EVALUATED ON ROTATIONAL DATASET. THE VALUES ARE EXPRESSED IN REC/PREC/F-MEASURE FORMAT. WE FORCE THE ACF AND ALMD-YOLO TO OUTPUT ONE BOUNDING BOX PER IMAGE. THUS, THE THREE VALUES ARE EQUAL

Method \ IoU	0.5	0.6	0.7
ACF [7]	74.8/74.8/74.8	60.9/60.9/60.9	39.6/39.6/39.6
Faster-rcnn [28] (VGG16)	87.6/85.9/86.8	69.0/67.7/68.4	50.8/49.8/50.3
YOLO-no angle [27]	85.7/94.2/89.7	68.5/75.3/71.7	40.7/44.8/42.6
SSD [19]	89.6/90.7/90.1	76.8/77.8/77.3	62.7/63.4/63.1
R-FCN [4]	83.9/84.6/84.3	74.0/74.6/74.3	60.9/61.4/61.2
MD-YOLO	92.6/94.4/93.5	80.5/82.1/81.3	60.0/61.2/60.6
MD-YOLO +data augment	95.6/97.3/96.4	84.3/85.8/85.0	58.4/59.5/58.9
ALMD-YOLO	98.4/98.4/98.4	92.7/92.7/92.7	75.2/75.2/75.2
ALMD-YOLO +data augment	**99.5/99.5/99.5**	**93.8/93.8/93.8**	**79.5/79.5/79.5**

It is worth noting that, when the postpositive model in ALMD-YOLO is trained with attention regions as input, it is preferable to create a new training set. That is, an image in the AOLP dataset is randomly cropped to the scale $(2 \times width) \times (3 \times height)$ relative to the ground truth. The image cropping attention-region training set is illustrated in Fig. 5.

B. Evaluation Criterion and Quantitative Comparison

In this subsection, we report comparisons of both the performance and computational complexity of our proposed method with those of other state-of-the-art methods. As there is no uniform criterion for evaluating the performance of different car license plate detection methods, we adopted the evaluation rule of general text detection, i.e., we employed precision, recall, and F-score measurements.

As regards the accuracy, different IoU thresholds of 0.5, 0.6, and 0.7 were adopted to verify the efficacy of our method. In Table IV, the detection accuracy of our proposed method is compared with those of existing state-of-the-art techniques, including R-FCN [4], ACF [7], SSD [19] and Faster-rcnn [28]; hence, it is evident that our proposed methodology outperforms the other existing techniques. Note that the YOLO-no angle entry indicates a case in which the same CNN was used as for MD-YOLO, but without rotation angle prediction.

From the Table IV, we find that our proposed MD-YOLO method exhibits superior performance to the other state-of-the-art methods, and that the data augmentation strategy also augments the results. Without the random rotation strategy, the rotational car license plates only occupy 1/3 of the dataset with some specific rotation angles. However, through this strategy, we enforce the rotation angle to uniformly distribute in interval $[-30°, +30°]$, which is able to improve the adaptability of model to varied rotation angles and then further improve the detection accuracy.

Obviously, without angle prediction, the YOLO-no angle model yields inferior performance to MD-YOLO; this reveals that our method manages the problem of multi-directional car license plate detection very well. Besides, with the introduction of attention-like prepositive CNN model, ALMD-YOLO performs better than the single MD-YOLO. Interestingly, the best performance of 79.5% is exhibited by the ALMD-YOLO framework, even when the IoU threshold is set to 0.7.

TABLE V

COMPARISON WITH OTHER CAR LICENSE PLATE DETECTION METHODS ON SUBSETS OF AOLP WITH IOU=0.5. THE VALUES ARE EXPRESSED IN PREC/REC FORMAT.

Method \ Subset	AC	LE	RP
Hsu et al. [12]	91/96	91/95	91/94
Li et al. [18]	98.53/98.38	97.75/97.62	95.28/95.58
ALMD-YOLO +data augment	**99.51/99.51**	**99.43/99.43**	**99.46/99.46**

In addition, we also present a comparison with other car license plate detection methods on each AOLP subset (Table V). Previously, Li and Shen [18] proposed a CNN-based method that adopts a sliding window strategy; however, this method is slower than our proposed technique, requiring a period of 2-3 s to run on GPU K40 (Table VII). Further, Hsu *et al.* [12] have proposed an edge-based method utilizing the EM algorithm. The results yielded by both of these algorithms are compared with our proposed method in Table V. Note that the results of Li and Shen [18] and Hsu *et al.* [12] are cited from [18] and both of them used the horizontal dataset. However, the rotational dataset is adopted for our method which is much harder to be detected than the horizontal dataset, revealed in Table III.

To prove the generality of our proposed method, we conducted the experiment on two other datasets, i.e. PKU (the Peking University) vehicle dataset [43] and UCSD (the University of California, San Diego) Calit2 dataset [17]. For PKU dataset [43], it contains 3977 vehicle images which were captured from various scenes (e.g. highways, intersections with crosswalks) under diverse conditions including daytime and nighttime. As for UCSD dataset [17], the vehicle images in dataset were taken by cameras positioned near roadways or parks. We use the images contained in its subset Gilman (1290 samples) and Stills (291 samples). Because the Gilman subset does not contain the ground truths of car license plates, we manually labeled the images in this subset. When splitting dataset into training set and test set, we employ ratio 2:1 for both of two datasets. Note that the car license plates in these two datasets are horizontal. To evaluate the performance of our method on multi-directional detection, we randomly rotated the images in training set and test set, within $[-30°, +30°]$. The aforementioned techniques R-FCN [4], ACF [7], SSD [19] and Faster-rcnn [28] were also applied on these two datasets to make a comparison with our method. From Table VI, it can be seen that our method (ALMD-YOLO) still outperforms others, which further demonstrates the generality of our method.

TABLE VI

F-MEASURE OF EACH METHOD ON UCSD AND PKU DATASETS. THE RESULTS WERE EVALUATED UNDER THE IOU THRESHOLD OF 0.5. OUR ALMD-YOLO METHOD HAS EMPLOYED THE SAME MODEL MENTIONED IN SECTION III WITH SAME DATA AUGMENTATION STRATEGY

Method \ Dataset	UCSD [17]	PKU [43]
ACF [7]	76.56	75.17
Faster-rcnn [28] (VGG16)	85.54	83.61
SSD [19]	85.32	86.63
R-FCN [4]	89.78	84.40
ALMD-YOLO (ours)	**98.32**	**97.38**

TABLE VII

COMPARISON OF COMPUTATIONAL COMPLEXITIES OF VARIOUS DETECTION METHODS ON GPU GTX980 AND CPU I7-2600. THE "-" SYMBOL INDICATES THAT THE COMPUTATIONAL PERIOD IS TOO LARGE TO REPORT OR THE METHOD CANNOT RUN IN THAT MODE

Method	GPU	CPU
Hsu et al. [12]	-	90ms
ACF [7]	-	**19.53ms**
Li et al. [18]	2.5s	-
Faster-rcnn [28] (VGG16)	175ms	-
SSD [19] (300*300)	50ms	-
R-FCN [4] (Resnet-50)	125ms	-
MD-YOLO	**5ms**	530ms
ALMD-YOLO	(5+2.7)ms	(530+140)ms

Furthermore, to validate the low computational complexity of our proposed method, all experiments were conducted on a single graphics processing unit (GPU) GTX980 and single central processing unit (CPU) i7-2600. A latency analysis of the different algorithms is presented in Table VII. It is apparent that our proposed method requires the shortest time on the GPU, exhibiting superior performance to all other methods. Further, although the methods reported in [7] and [12] exhibit faster than ours on the CPU, the accuracy of our proposed model is significantly higher.

Overall, to the best of our knowledge, our proposed method yields the highest detection accuracy for car license plate detection methods reported till-date, while simultaneously being feasibly applicable to real-time car license plate detection scenarios. Examples of car license plates detection by our proposed method are presented in Fig. 9.

C. Analysis

Aforementioned quantitative results demonstrate that our proposed method significantly outperforms previous methods in literature. Generally speaking, such improved performance has obtained due to the precise prediction of the rotation angle of multi-directions. The task of multi-directional detection is considered to be more challenging than the general objects detection. Once the predicted angle deviates from the true one, the overlap between predicted bounding box and ground truth will drop dramatically. The previous detection frameworks [4], [19], [27], [28] perform well on the general objects detection. However, the performance of such algorithms is severely deteriorated in terms of horizontal minimum rectangles to enclose the rotated objects, especially for the multi-directional car license plate detection. Such kind of detection results inevitably introduce redundant portions and background noise in the rectangular region, which is adverse to the subsequent car license plate recognition. Fig. 6 explains this phenomenon and draws a decreasing curve of IoU. With the increase of rotation angle, the general objects detectors predict bigger circumscribed rectangle, trying to enclose all the portions of rotational object. Meanwhile, the redundant portions have also been extended, which finally causes the decrease in IoU. Specifically, in Fig. 7, we present

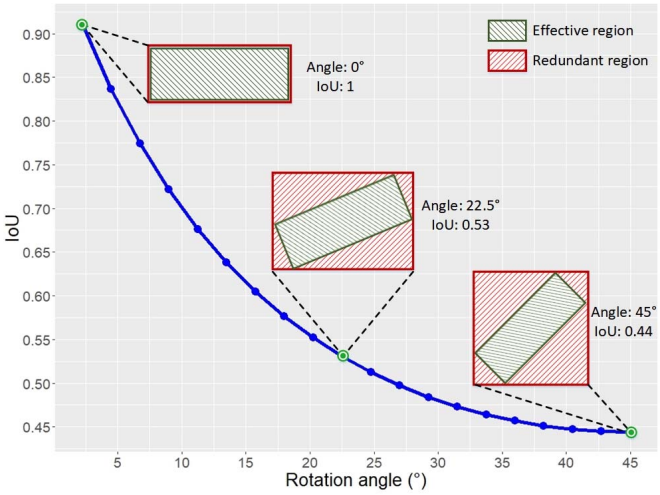

Fig. 6. The decreasing curve of IoU w.r.t. rotation angle. Increasing rotation angle dramatically decreases the IoU between circumscribed rectangle and rotational rectangle. Meanwhile, the general objects detectors will detect larger redundant region.

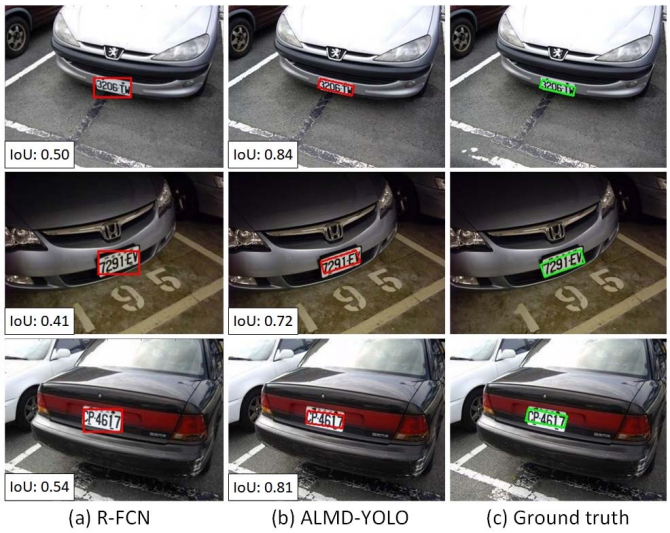

Fig. 7. Comparison between R-FCN and ALMD-YOLO. With precise rotation angle prediction, our detection method is able to locate the car license plate tightly.

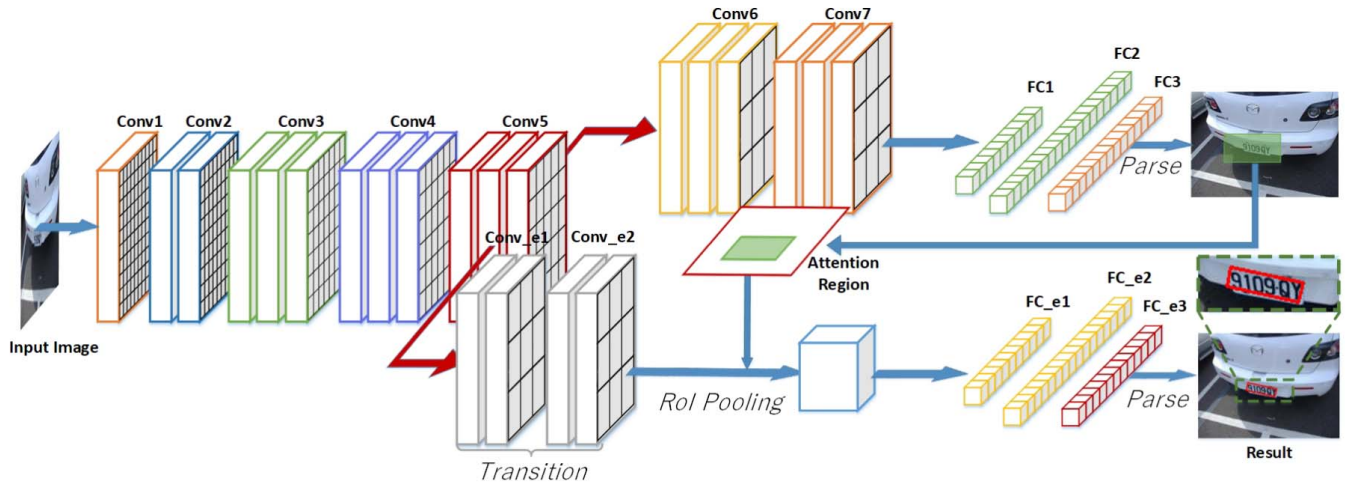

Fig. 8. An end-to-end model which has employed the same regression mechanism as ALMD-YOLO.

Fig. 9. Examples of car license plate detection.

a comparison with R-FCN [4] on some typical samples. With the precise rotation angle prediction, our detection method is able to locate the car license plate tightly.

D. Comparison Between ALMD-YOLO and an End-to-End Model

In order to show our model with two-stage training strategy outperforms the corresponding end-to-end one, we design an end-to-end model by the same rules as ALMD-YOLO. Fig. 8 illustrates the architecture of the end-to-end network. For comparison, the prepositive CNN network introduced in Section III is adopted here as backbone (conv1-conv5) which is responsible for extracting features and providing the shared features for subsequent subnets. When sharing the feature maps of conv5, two extra convolutional layers are inserted as transition layers to balance two branches. To ensure that the fully connected layer of second subnet receives the inputs of same shape, we employ the RoI pooling layer which is proposed in [9]. Besides, the data augmentation strategy, the fully connected layers and detection mechanism are consistent with ALMD-YOLO. Table VIII presents the comparison

TABLE VIII

THE F-SCORE OF AN END-TO-END MODEL AND SPLIT-TRAINING MODEL (ALMD-YOLO)

Method \ IoU	0.5	0.6	0.7
End-to-end	98.6	90.1	61.8
ALMD-YOLO	**99.5**	**93.8**	**79.5**

between the two-stage model (ALMD-YOLO) and the end-to-end model. Apparently, there is a distinct gap between two models, especially in the situations of IoU = 0.6 and IoU = 0.7. The two-stage model achieves higher detection accuracy. In fact, it is relatively difficult to integrate two subnets in an end-to-end model as we should guarantee that the second subnet receives the inputs of the same shape. Many contemporary methods for end-to-end detection, e.g. Faster-rcnn [28], R-FCN [4], employ special pooling operations including RoI pooling [9], position-sensitive pooling [4] to acquire the outputs with fixed shape. Such special pooling operations sacrifice the features to some extent, which is harmful to multi-directional detection. In contrast, the MD-YOLO receives the portion of raw image from prepositive CNN without any information loss so that the inputs can retain the rotation information perfectly. We found that the end-to-end model merely saves the time consumption of the second network of ALMD-YOLO. From Table VIII, it can be seen that our two-stage ALMD-YOLO model is much better than the end-to-end model.

V. CONCLUSION

In this paper, we have introduced a new MD-YOLO model for multi-directional car license plate detection. The proposed model can elegantly solve the problem of multi-directional car license plate detection, and can also be deployed easily in real-time circumstances, because of its reduced computational complexity compared with previous CNN based methods. With the introduction of an attention-like prepositive CNN model, the ALMD-YOLO framework yields new state-of-the-art accuracy. Moreover, the multi-directional car license plate detection method presented in this paper can handle challenging real-world scenarios reasonably well, even in cases when only limited computational resources, such as CPUs, are available. However, there are still couple of issues for robust license plate detection, which need to be addressed in future. Firstly, The lack of enough data of multi-directional car license plates limits the performance of our method, which will force us to collect more data or employ some advanced techniques to synthesize data. Besides, although we focus on the multi-directional car license plate detection in this paper, the detection still aims for the subsequent recognition. Therefore, it deserves to further explore whether we can propose an end-to-end model to simultaneously deal with multi-directional detection and recognition. Last but not least, the tasks of multi-directional car license plate detection under poor conditions such as low resolution, terrible illumination, accidental occlusion and so on remain great challenges waiting to be solved.

REFERENCES

[1] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1690–1705, Aug. 2014.

[2] R. Azad, F. Davami, and B. Azad, "A novel and robust method for automatic license plate recognition system based on pattern recognition," *Adv. Comput. Sci., Int. J.*, vol. 2, no. 3, pp. 64–70, 2013.

[3] F. A. da Silva, A. O. Artero, M. S. V. de Paiva, and R. L. Barbosa. (2013). "ALPRS—A new approach for license plate recognition using the sift algorithm." [Online]. Available: https://arxiv.org/abs/1303.1667

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[5] K. Deb, H.-U. Chae, and K.-H. Jo, "Vehicle license plate detection method based on sliding concentric windows and histogram," *JCP*, vol. 4, no. 8, pp. 771–777, 2009.

[6] L. Dlagnekov, "License plate detection using adaboost," Dept. Comput. Sci. Eng., San Diego, CA, USA, Tech. Rep., 2004. [Online]. Available: https://cseweb.ucsd.edu/classes/fa04/cse252c/projects/louka.pdf

[7] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[8] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, Feb. 2013.

[9] R. Girshick, "Fast R-CNN," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[10] W. T. Ho, H. W. Lim, and Y. H. Tay, "Two-stage license plate detection using gentle adaboost and SIFT-SVM," in *Proc. 1st Asian Conf. Intell. Inf. Database Syst. (ACIIDS)*, Apr. 2009, pp. 109–114.

[11] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. 2004, pp. 831–834.

[12] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 552–561, Feb. 2013.

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[14] H. Karwal and A. Girdhar, "Vehicle number plate detection system for indian vehicles," in *Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol. (CICT)*, Feb. 2015, pp. 8–12.

[15] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park. (2016). "PVANET: Deep but lightweight neural networks for real-time object detection." [Online]. Available: https://arxiv.org/abs/1608.08021

[16] S. Kim, D. Kim, Y. Ryu, and G. Kim, "A robust license-plate extraction method under complex image conditions," in *Object Recognition Supported by User Interaction for Service Robots*, vol. 3. Quebec City, QC, Canada: IEEE Press, 2002, pp. 216–219.

[17] L. Dlagnekov and S. Belongie. (2005). *UCSD/CALIT2 Car License Plate, Make and Model Database*. [Online]. Available: http://vision.ucsd.edu/car_data.html

[18] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMS," *CoRR*, Jan. 2016. [Online]. Available: https://arxiv.org/abs/1601.05610

[19] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[20] D. C. Luvizon, B. T. Nassu, and R. Minetto, "Vehicle speed estimation by license plate detection and tracking," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2014, pp. 6563–6567.

[21] F. Martín, O. Martín, M. García, and J. L. Alba, "New methods for automatic reading of VLP's (vehicle license plates)," in *Proc. IASTED Int. Conf. SPPRA*, 2002, pp. 126–131.

[22] J. Muhammad and H. Altun, "Improved license plate detection using HOG-based features and genetic algorithm," in *Proc. 24th Signal Process. Commun. Appl. Conf. (SIU)*, May 2016, pp. 1269–1272.

[23] C. Patel, A. Patel, and D. Shah, "A novel approach for detecting number plate based on overlapping window and region clustering for Indian conditions," *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 5, pp. 58–65, Apr. 2015.

[24] The CGAL Project. (2016). *The Computational Geometry Algorithms Library*. [Online]. Available: http://www.cgal.org

[25] A. Rabee and I. Barhumi, "License plate detection and recognition in complex scenes using mathematical morphology and support vector machines," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2014, pp. 59–62.

[26] S. A. Radzi and M. Khalil-Hani, "Character recognition of license plate number using convolutional neural network," in *Visual Informatics: Sustaining Research and Innovations* (Lecture Notes in Computer Science), vol 7066, H. Z. Badioze *et al.*, Eds. Berlin, Germany: Springer, 2011, pp. 45–55.
[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[29] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[30] Z. Saidane and C. Garcia, "Automatic scene text recognition using a convolutional neural network," in *Proc. Workshop Camera-Based Document Anal. Recognit.*, vol. 1. 2007, pp. 100–106.
[31] H. Samma, C. P. Lim, J. M. Saleh, and S. A. Suandi, "A memetic-based fuzzy support vector machine model and its application to license plate recognition," *Memetic Comput.*, vol. 8, no. 3, pp. 235–251, 2016.
[32] Q. Wang, J. Gao, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 219–224, 2018.
[33] Q. Wang, J. Gao, and Y. Yuan, "A joint convolutional neural networks and context transfer for street scenes labeling," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2017.2726546.
[34] Q. Wang, Y. Yuan, P. Yan, and X. Li, "Saliency detection by multiple-instance learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 660–672, Apr. 2013.
[35] Q. Wang, G. Zhu, and Y. Yuan, "Multi-spectral dataset and its application in saliency detection," *Comput. Vis. Image Underst.*, vol. 117, no. 12, pp. 1748–1754, Dec. 2013.
[36] R. Wang, N. Sang, R. Wang, and L. Jiang, "Detection and tracking strategy for license plate detection in video," *Opt.-Int. J. Light Electron Opt.*, vol. 125, no. 10, pp. 2283–2288, 2014.
[37] S.-Z. Wang and H.-J. Lee, "Detection and recognition of license plate characters with different appearances," in *Proc. IEEE Intell. Transp. Syst.*, vol. 2. Oct. 2003, pp. 979–984.
[38] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 3304–3308.
[39] D. Wazalwar *et al.*, "A design flow for robust license plate localization and recognition in complex scenes," *J. Transp. Technol.*, vol. 2, no. 1, p. 13, 2012.
[40] M. S. Yee and L. Hanzo, "Radial basis function decision feedback equaliser assisted burst-by-burst adaptive modulation," in *Proc. Global Telecommun. Conf.*, vol. 4. 1999, pp. 2183–2187.
[41] K. M. A. Yousef, M. Al-Tabanjah, E. Hudaib, and M. Ikrai, "Sift based automatic number plate recognition," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 124–129.
[42] Y. Yuan, Z. Xiong, and Q. Wang, "An incremental framework for video-based traffic sign detection, tracking, and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1918–1929, Jul. 2017.
[43] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A robust and efficient approach to license plate detection," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1102–1114, Mar. 2017.
[44] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, 2005.

Lele Xie is currently pursuing the master's degree in electronic and information engineering from South China University of Technology, China. His major research interests include object detection, text detection, image processing, and deep learning.

Tasweer Ahmad received the bachelor's degree in electrical engineering from University of Engineering and Technology, Taxila, Pakistan, in 2007, and the master's degree in electronic and communication engineering from University of Engineering and Technology, Lahore, Pakistan, in 2009. He is currently pursuing the Ph.D. degree from South China University of Technology, China.

He was an Instructor with the Government College University, Lahore, from 2010 to 2015, and with the COMSATS Institute of Information Technology, Sahiwal Campus, Pakistan, from 2015 to 2016. His current research interests include image processing, computer vision, and machine learning.

Lianwen Jin (M'98) received the B.S. degree from University of Science and Technology of China, Anhui, China, and the Ph.D. degree from South China University of Technology, Guangzhou, China, in 1991 and 1996, respectively. He is currently a Professor with the College of Electronic and Information Engineering, South China University of Technology. His research interests include handwriting analysis and recognition, image processing, machine learning, and intelligent systems.

He has authored over 100 scientific papers. He has received the New Century Excellent Talent Program of MOE Award and the Guangdong Pearl River Distinguished Professor Award. He is a member of the IEEE Computational Intelligence Society, IEEE Signal Processing Society, and IEEE Computer Society.

Yuliang Liu received the B.S. degree from South China University of Technology in 2016, where he is currently pursuing the Ph.D. degree. His current research interests include deep learning and object detection and text recognition.

Sheng Zhang received the B.S. degree from Hohai University, Nanjing, China, in 2014. He is currently pursuing the Ph.D. degree with the HCII Laboratory, South China university of Technology. His current research interests include object tracking, deep learning, and image processing algorithms.