# Simultaneous End-to-End Vehicle and License Plate Detection With Multi-Branch Attention Neural Network

Song-Lu Chen, Chun Yang, Jia-Wei Ma, Feng Chen, and Xu-Cheng Yin, *Senior Member, IEEE*

***Abstract*— Vehicle and license plate detection plays an important role in intelligent transportation systems and is still a challenging task in real applications, such as on-road scenarios. Recently, Convolutional Neural Network (CNN)-based detectors achieve the state-of-the-art performance. However, it is difficult to efficiently detect the vehicle and license plate simultaneously in most cases. With a single network, the vehicle can affect the detection of the license plate due to the inclusion relation. In this paper, we propose an end-to-end deep neural network for detecting the vehicle and the license plate simultaneously in a given image, where two separate branches with different convolutional layers are designed for vehicle detection and license plate detection, respectively. In consideration of the license plate's small size and fairly obvious features as well as the vehicle's various size and rather complex features, the license plates are detected with low-level features and the vehicles are localized with multi-level features in corresponding convolutional layers. Moreover, a task-specific anchor design strategy is employed to obtain better predictions. Besides, the attention mechanisms and feature-fusion strategies are utilized to improve the detection performance of small-scale objects. A variety of experiments on real datasets and public datasets verify that our proposed method has fairly high accuracy and efficiency.**

***Index Terms*— Vehicle detection, license plate detection, end-to-end, multi-branch, attention.**

## I. Introduction

AUTOMATIC vehicle and license plate detection are important in intelligent transportation systems. A variety of methods have been proposed in the literature. However, license plate detection is still considered as a challenging task in real applications because of the small size of captured license plates, illumination variations in the scene and viewpoint changes of cameras. Similarly, the problem of vehicle detection in real scenes is also unsolved because of vehicle size changes, vehicle poses variations and complex scene backgrounds.

Before the deep learning era, most object detection methods need to specifically design hand-engineered features for different objects. For the vehicle, most detection approaches [1] usually utilized information about symmetry, color, shadow, geometrical features (e.g., corners, horizontal/vertical edges), texture features and vehicle lights. As for the license plate, the available detection methods [2] can be roughly classified into five categories: edge-based, connectivity-based, texture-based, color-based and character-based methods. Recently, the deep Convolutional Neural Networks (CNNs) can learn features automatically from a large amount of training data. In [3], [4], CNN-based methods are utilized to detect the vehicle only, and [5]–[7] are proposed to detect the license plate directly. Moreover, some methods use a cascaded strategy to detect the vehicle and the license plate, where vehicles are firstly detected, and the license plate is correspondingly localized in each vehicle region [8]–[10]. However, the above-mentioned methods either regard vehicle detection and license plate detection as two independent tasks, or detect the license plate in cascaded ways, which are less efficient. Moreover, in a cascaded way, the detection of the license plate depends on the quality of the vehicle proposals, and it is certain to be failed if the corresponding vehicle is not detected. One better way is to detect the vehicle and the license plate simultaneously as a multi-task learning system.

There have been several powerful object detection methods, e.g., SSD [11], YOLO [12] and Faster R-CNN [13]. However, we find it difficult to detect the vehicle and license plate simultaneously using these prestigious frameworks. As seen in Figure 1, some distinct license plates are unexpectedly failed to be detected, and the confidence of detected ones is also at a low level. In deep neural networks, the vehicle and the license plate generally share the same head networks and anchor boxes. Thus, license plate detection is easily affected by the vehicle because of their inclusion relation. In this paper, we propose an end-to-end multi-branch attention neural network for simultaneously detecting the vehicle and the license plate in a given image, where two ***separate*** branches with different convolutional layers are stemmed from the backbone network to detect the vehicle and the license plate respectively. In general, the low-level features of CNNs have high resolution with weak semantics and are important to

Manuscript received May 20, 2019; revised July 24, 2019; accepted July 25, 2019. Date of publication August 6, 2019; date of current version August 28, 2020. This work was supported in part by the Beijing Natural Science Foundation under Grant 4194084, in part by the China Postdoctoral Science Foundation under Grant 2018M641199, and in part by the Fundamental Research Funds for the Central Universities under Grant FRF-TP-18-060A1. The Associate Editor for this paper was W. Lin. *(Song-Lu Chen and Chun Yang contributed equally to this work.) (Corresponding author: Xu-Cheng Yin.)*

S.-L. Chen, C. Yang, J.-W. Ma, and X.-C. Yin are with the Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing 100083, China, and also with the USTB-EEasyTech Joint Laboratory of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China (e-mail: chenslvs7@gmail.com; chunyang@ustb.edu.cn; mjw20151001@hotmail.com; xuchengyin@ustb.edu.cn).

F. Chen is with EEasy Technology Company Ltd., Zhuhai 519000, China (e-mail: chenfengzh@126.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Digital Object Identifier 10.1109/TITS.2019.2931791

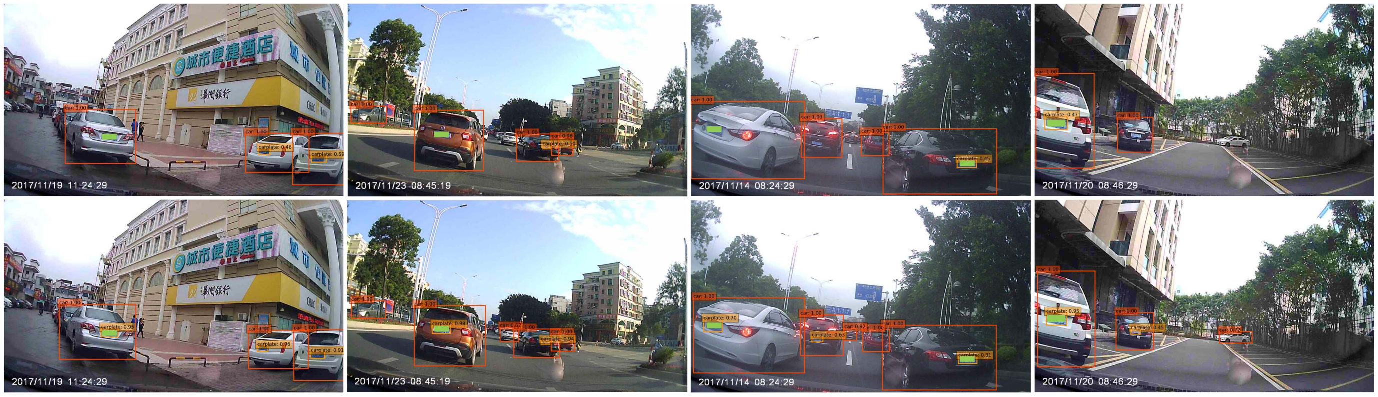

Fig. 1. Examples from SSD and our method, first row for SSD and second row for ours. With SSD, some small-scale vehicles and license plates are not detected, and the confidence of detected license plates is at a low level. All recognizable license plates are manually blurred to protect the privacy.

small object detection. Meanwhile, high-level features are semantically strong but with low-resolution, and these features have better feature representation of large objects [14], [15]. Intuitively, the license plates usually have a relatively small size and fairly simple features, while the vehicles have various scales and rather complex features. Accordingly, we assign low-level features for license plate detection and multi-level features for vehicle localization.

Moreover, a task-specific anchor design strategy is also applied for vehicle and license plate detection. For both two kinds of objects, we select better anchor priors instead of hand-picked ones based on the clustering method [16], [17], which can make it easier to learn better predictions, as detailed in Section III-C. Besides, as shown in Figure 1, small objects are always failed to be detected, especially for the license plates and vehicles in the distance. Inspired by [18], we add the spatial attention mechanisms into each branch to facilitate focusing on the regions of interest (ROIs). Additionally, we apply the feature-fusion strategy of combining both high-resolution, semantically weak features and low-resolution, semantically strong features to leverage the pyramidal feature hierarchy of CNNs, as detailed in Section III-D.

In summary, our paper has three main contributions.

- We propose an end-to-end multi-branch attention neural network for detecting the vehicle and the license plate simultaneously, which has two separate branches with different convolutional layers for vehicle detection and license plate detection respectively. In this way, the vehicle's effects on the license plate are eliminated.
- We collect three large-scale datasets with annotating both vehicles and license plates, where one is a private dataset and the other two are re-annotated from the public datasets. Supplementary materials and two re-annotated datasets are now available at ***https://github.com/chensonglu/Vehicle_License_Plate_Datasets***.
- we employ the attention mechanisms and feature-fusion strategies to enhance the detection of the small-scale objects. Moreover, we apply a task-specific anchor design strategy to generate better predictions for vehicle and license plate detection. Finally, extensive experiments validate the effectiveness and efficiency of our method.

The rest of this paper is organized as follows. Related work is described in Section II. In Section III, we describe our method in details. Section IV presents the comparative experiments. Final remarks are presented in Section V.

## II. Related Work

Vehicle and license plate detection has drawn considerable research attention. Previously, most methods often extracted hand-engineered features, such as texture features and edge features, for object detection. Recently, people usually utilize Deep Neural Networks (DNNs) for feature representation.

**CNNs for Object Detection** Over the past few decades, methods with economic features and inference schemes have been popular for efficiency, such as DPM [19]. In recent years, the DNNs have been driving the advance of object detection due to the powerful ability of feature representation, and the CNN-based approaches have achieved state-of-the-art performances. R-CNN [20] is a milestone for object detection, which utilizes Selective Search [21] to generate excessive region proposals and then apply CNNs to classify each region. The follow-up Faster R-CNN [13] proposes the region proposal network (RPN), and combines it with the detection block [22] into an end-to-end detection framework with two stages. Moreover, YOLO [12] and SSD [11] can directly predict/regress object bounding boxes using an end-to-end network in a single shot, where SSD [11] can detect the object of various scales by combining multi-scale features. YOLOv2 [16] proposes a dimension clustering strategy to automatically find better priors for better detections. FPN [23] attempts to create feature pyramids that have strong semantics at all scales by combining low-level features and high-level features. FAN [18] utilizes an attention mechanism to improve the detection of the occluded faces [24].

**Vehicle Detection** [25] utilizes hand-engineered texture features for vehicle detection. [3] optimizes a CNN architecture for vehicle detection under different weather conditions. [4] applies a SSD-based network to detect vehicles on the expressway, and mainly focuses on the detection of small-scale and motion-blurred vehicles.

**Direct License Plate Detection** [26] proposes a novel method to detect the license plate by principal visual word, discovery and local feature matching, which can adaptively cope with different changes of the license plate, such as rotation, scaling, illumination. Reference [27] presents a robust and efficient approach for license plate detection, which firstly accelerates the license plate localization using an effective image down-scaling method, and then utilizes dense filters to extract candidate regions, and finally identifies the true license plates using a cascaded classifier. Reference [5] utilizes customized YOLO [12] and YOLOv2 [16] to handle license plate detection in the wild, which deals with the license plates captured under conditions like bad weathers, lighting, traffics. Reference [6] presents a method for license plate detection aiming at images captured with low-resolution cameras from a long distance. Reference [7] proposes a CNN-based MD-YOLO framework for multi-directional license plate detection.

**Cascaded License Plate Detection** [8] proposes a method for license plate detection using vehicle region extraction, which utilizes R-CNN [20] to generate vehicle proposals and then localize the license plate in each vehicle region. Reference [9] proposes a cascaded convolutional neural network for license plate detection, which firstly applies the RPN module to generate candidate vehicle proposals and then detects the license plate based on each proposal. Reference [10] introduces a novel CNN framework capable of detecting and rectifying multi-directional license plates in a cascaded way.

## III. Methodology

We propose an end-to-end multi-branch attention neural network to detect vehicle and license plate simultaneously, where two separate branches with different convolutional layers are designed for vehicle detection and license plate detection respectively. The license plates are detected with low-level features and the vehicles are localized with multi-level features. Moreover, a task-specific anchor design strategy is applied for better object predictions. Besides, we employ the attention mechanisms and feature-fusion strategies to improve the recall of small-scale cases. The overall network architecture is demonstrated in Figure 2. Note: The depth of the attention masks, classification, and regression head layers is 1, and Figure 2 is only for demonstration.

### A. Base Network

The backbone network is inherited from the popular VGG-16 [28], keeping convolutional layers from conv1_1 to conv5_3. The last two fully-connected layers (fc6, fc7) are converted into convolutional layers and extra layers from conv6_2 to conv9_2 are also added for semantically stronger feature extraction, which is the same with SSD [11] for a fair comparison. The details of the backbone network are described in the supplementary materials.

### B. Detection Branch

It is difficult to detect the vehicle and license plate simultaneously using the same head networks (classifier and regressor) and anchor boxes because of the inclusion relation. The vehicle can easily affect the detection of the license plate when these two objects act on the same anchor boxes through a shared classifier and regressor. Thus, it is necessary to decouple the relationships between the two objects, and vehicle and license plate detection can be separated into different branches. Each branch holds its head network for object classification and bounding box regression respectively. Moreover, low-level features can be useful for license plate detection due to the small size and fairly obvious features, and multi-level features can be used for vehicle localization due to the various scales and rather complex features. As illustrated in Figure 2, we assign several relatively shallow layers to the license plate detection branch and allocate dispersed layers to the vehicle detection branch. Moreover, the scale of features in different layers may be quite different, making it difficult to combine them for detection directly. The shallow features need to be normalized before combining with the deep features to avoid parameter imbalance. More details of normalization are described in the supplementary materials.

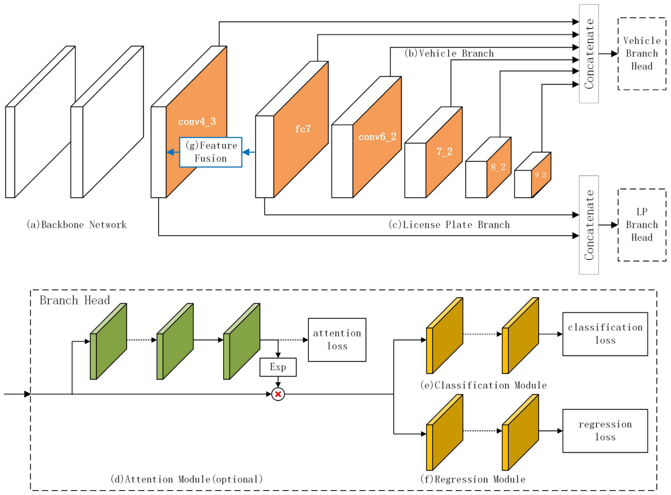

Fig. 2. Network architecture. The backbone network is inherited from VGG-16 with four extra layers. The license plates are detected with low-level features, and the vehicles are localized with multi-level features, where conv4_3 and fc7 are shared. Each detection branch holds its head layers for classification and bounding box regression, where the attention module is adopted at low-level features to highlight the foreground information. Moreover, the feature fusion module is applied to enhance the semantics of the lower layers in each detection branch.

### C. Anchor Design Strategy

The performance of the sliding-window based methods largely depends on the selection strategy of the anchor boxes. Even the network can learn to adjust the boxes appropriately, better anchor priors make it easier to predict better detection and make the network converge faster. However, with SSD [11], the scale and aspect ratio of the anchor boxes are all set manually according to empirical experience. Under this mode, it is difficult to cover objects with uncommon scales and aspect ratios, such as license plates and vehicles in the distance. Inspired by [16], [17], one can automatically obtain more suitable scales and aspect ratios of the anchor boxes by using anchor clustering. Firstly, we run K-means clustering to get the anchor priors based on YOLOv2 [16], where the distance metric is shown in (1) (GT means

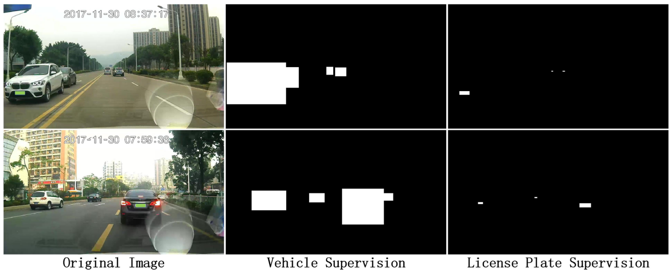

Fig. 3. Attention supervision. The vehicles and license plates are filled with 1 respectively for each detection branch, and the background is filled with 0. The first column shows the original images. The second and third columns demonstrate the filled ground truths of vehicle and license plate respectively. All recognizable license plates are manually blurred to protect the privacy.

ground truth). Then, all priors are sorted by area from small to large. Finally, the sorted priors are allocated to different layers like YOLOv3 [17], where small-scale anchors are placed on the low-level feature maps and large-scale anchors are placed on the high-level feature maps. The center of each anchor box is set to $\left(\frac{i+0.5}{|s_k|}, \frac{j+0.5}{|s_k|}\right)$, where $|s_k|$ is the size of the k-th feature map, $i, j \in [0, |s_k|)$.

$$d(GT_{box}, centroid) = 1 - IoU(GT_{box}, centroid) \quad (1)$$

Moreover, the average IoU [16] can be calculated with the closest centroid without considering the spatial position of the anchor boxes. The average IoU is computed under the ideal conditions. However, anchor boxes of SSD-based [11] methods are scattered in a sparse way, where the average IoU should be calculated with the spatial anchor boxes and we call it spatial IoU. More details about the anchor clustering are provided in the supplementary materials.

### *D. Attention and Feature Fusion*

According to Figure 1, in real scenes, it is challenging to detect small-scale objects, especially for the license plates and vehicles in the distance. As mentioned in [18], the spatial attention mechanism can highlight the foreground information and keep the context information. We add a segmentation-like mask before the classification and regression module. Only the two shallowest layers in each detection branch adopt the attention module because deeper layers have large receptive fields and easily bring in noises. For both branches, the attention module helps to highlight features of the foreground regions and diminish the background regions. The attention supervision information is simply obtained by filling the ground truth as shown in Figure 3 and the attention loss is simply pixel-wise sigmoid cross-entropy between the filled ground truth and the predicted mask. Finally, the attention maps are fed into exponential operation and then have dot product with the feature maps.

Figure 4 demonstrates two examples of the predicted attention mask. As can be seen, it tends to focus on the center of objects. Moreover, the attention mask of high-level feature maps can cover more RoIs because of larger respective fields. In each detection branch, the attention module helps to enhance the foreground information, which is favorable to small-scale objects, as shown in Section IV-E.

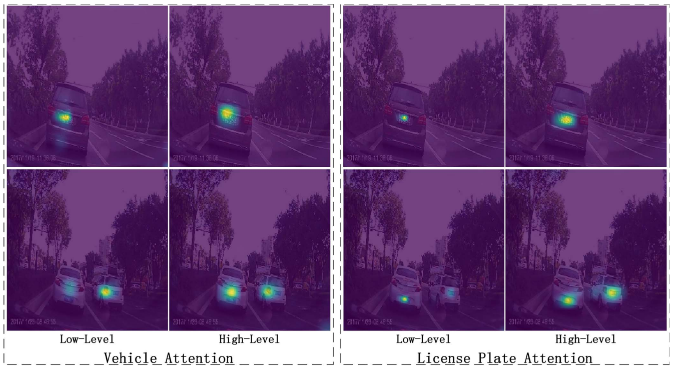

Fig. 4. Attention mask. Each row shows one original image resized to 300*300, covered with the attention mask. The first two columns demonstrate the attention masks of the vehicle, and the last two columns demonstrate the attention masks of the license plate.

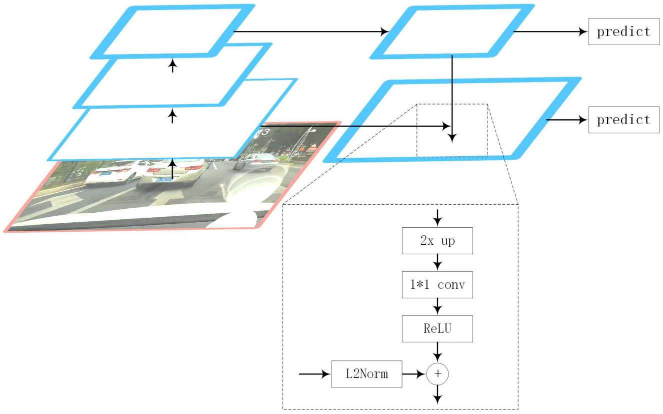

Fig. 5. Feature fusion building block illustrating the lateral connection and the top-down pathway.

Furthermore, fusing high-level features with low-level features [23], [29], [30] can enhance the semantic representation. ION [29] achieves feature fusion by simple concatenation, while FPN [23] proposes using element-wise addition. FSSD [30] proposes concatenating features together to generate a series of pyramid features for object detection. To further reduce computational complexity, we simply utilize the feature fusion strategy in the two shallowest layers of each detection branch, as illustrated in Figure 2. To achieve the speed-accuracy tradeoff, we adopt FPN [23] as the feature fusion module. The upper layer is firstly up-sampled by a factor 2 using nearest-neighbor interpolation, and then undergo a 1*1 convolutional layer to reduce channel dimensions. Considering the different scales between different layers shown in Section III-B, the upper layer needs to be rectified by ReLU and then merged with the normalized low-level features by element-wise addition. Figure 5 illustrates the building block of merging operation between lateral connection and top-down pathway. ION [29] and FSSD [30] are illustrated in the supplementary materials.

### *E. Training Objective*

The optimization function is composed of three parts. For the classification regression module, we adopt the same loss function as SSD [11]. Let $c$ be the confidence, $l$ be the

$$L(x,c,l,g,m) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) + \beta \sum_{k\in K}^{K} L_a(x, m_a^k, m_g^k) \tag{2}$$

$$L_{total}(x,c,l,g,m) = L_{Vehicle}(x,c,l,g,m) + \gamma L_{LP}(x,c,l,g,m) \tag{3}$$

predicted box, $g$ be the ground truth box, $N$ be the number of matched anchor boxes. For the attention module, we calculate the pixel-wise sigmoid cross-entropy between the generated attention mask and the ground truth. Let $K$ be the index of all used pyramidal features, $m_a^k$ be the attention mask generated per level, $m_g^k$ be the ground truth described in Figure 3. $\alpha$ and $\beta$ are the weighting parameters to balance these terms. We utilize two separate optimization functions for the vehicle detection branch and the license plate detection branch respectively. These two branches undergo the separate back-propagation process, and $\gamma$ is the weighting factor to adjust two branches. The loss function is defined as (2) and (3), shown at the top of this page, and we simply set $\alpha = \gamma = 1$ and $\beta = 3$.

We adopt the smooth L1 loss [13] and the softmax loss for regression and classification respectively and employ the pixel-wise sigmoid cross-entropy for attention loss. Let $K_t = 2$ be the class number when training vehicle and license plate together with shared head layers, $C_t = \{Vehicle, LP, Background\}$ be the classes. With SSD [11], the confidence of the n-th object is calculated by (4).

$$conf_t^n = \frac{e^{C_t^n}}{\sum_{m=1}^{K_t+1} e^{C_t^m}} \tag{4}$$

Let $K_V = 1$ be the class number of the vehicle detection branch, $C_V = \{Vehicle, Background\}$ be the classes. Let $K_{LP} = 1$ be the class number of the license plate detection branch, $C_{LP} = \{LP, Background\}$ be the classes. With our method, the confidence of the n-th object is calculated separately for each detection branch by (5) and (6).

$$conf_V^n = \frac{e^{C_V^n}}{\sum_{m=1}^{K_V+1} e^{C_V^m}} \tag{5}$$

$$conf_{LP}^n = \frac{e^{C_{LP}^n}}{\sum_{m=1}^{K_{LP}+1} e^{C_{LP}^m}} \tag{6}$$

As for the regression module, it predicts the offsets of position and scale to the anchor boxes. Suppose a anchor box $d = (x_d, y_d, w_d, h_d)$ and the predicted values $(\Delta x, \Delta y, \Delta w, \Delta h)$, the box $b = (x, y, w, h)$ can be obtained as follows.

$$x = x_d + w_d \Delta x \tag{7}$$

$$y = y_d + w_d \Delta y \tag{8}$$

$$w = w_d exp(\Delta w) \tag{9}$$

$$h = h_d exp(\Delta h) \tag{10}$$

## IV. Experiments

We adopt VGG-16 [28] as the base model, which is pre-trained on the ILSVRC CLS-LOC dataset [31]. The baseline network follows SSD300 [11]. All the training images are augmented with random crop and distortion, etc, following the same scheme as [11]. Our model is trained with 300 × 300 images using Adam [32] for 40k iterations. The momentum parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Learning rate, weight decay and batch size are set to $10^{-4}$, $5 \times 10^{-4}$, 32 respectively. All the experiments are carried on a PC with 4 NVIDIA TITAN Xp GPU.

### A. Datasets

*1) Datasets With Vehicles and License Plates:* Datasets are required to be labeled with the position coordinates of the vehicles and the license plates. Details about data collection and annotation are described in the supplementary materials.

**VALID** We employ two auto-mobile data recorders to collect videos on the road of a Chinese city[1] with the resolution of 720 × 1280. For simplicity, we name our dataset VALID (Vehicle And LIcense plate Dataset). A total of 887 images are collected and carefully annotated. 78 images from one recorder are used as the test set. The rest 809 images from another recorder are randomly divided into the training set and the validation set by 7:3.

**DETROIT** We re-annotate a subset from Open Image Dataset (OID) V4 [33], which contains "Car" and "Vehicle registration plate". For simplicity, we call it DETROIT (DatasET fRom Open Image daTaset). The images of DETROIT are obtained from the Internet, and the size and aspect ratio varies greatly. 386 images from the OID validation set are used as the test set. 1113 images from OID test set are randomly divided into the training set and the validation set by 7:3

**DOC** We combine vehicle position form Cars [34] and license plate position from [10] to obtain DOC (Dataset frOm Cars). A total of 105 images are obtained. 70% are randomly selected as the training-validation set, and the rest 30% is used as the test set. The images of DOC are also obtained from the Internet, and the size and aspect ratio varies greatly.

*2) Datasets With Vehicles or License Plates:* Moreover, to verify the effectiveness of each detection branch, datasets with annotations of only vehicles or license plates are needed.

**Udacity Dataset 1** Udacity self-driving dataset 1[2] contains over 65000 labels across 9420 frames collected from the cityscape. All labels of "Car" and "Truck" in Udacity Dataset 1 are transformed into "Vehicle", and all images are randomly divided into the training-validation set and test set by 7:3.

**AOLP-LE** AOLP-LE [35] was collected in the on-road scenario for license plate detection and recognition. All

[1] Zhuhai, China

[2] https://github.com/udacity/self-driving-car/tree/master/annotations

TABLE I

DETECTION RESULTS (%) OF TRAINING VEHICLE AND LICENSE PLATE SEPARATELY OR SIMULTANEOUSLY ON THE TEST SET OF VALID

| Method(Dataset) | $AP_{0.5}$ of Vehicle | $AP_{0.5}$ of LP |
|---|---|---|
| SSD(Vehicle only) | **83.96** | - |
| SSD(LP only) | - | **85.9** |
| SSD(Vehicle+LP) | 83.77 | 71.86 |

TABLE II

DETECTION RESULTS (%) ON THE PROCESSED TEST SET (FIGURE 6) OF VALID WITH SSD300 AND TWO BRANCHES (TB MEANS TWO BRANCHES AND DIS LP MEANS DISTRIBUTED LICENSE PLATES.)

| Method(Dataset) | $AP_{0.5}$ of Vehicle | $AP_{0.5}$ of LP |
|---|---|---|
| SSD(Vehicle w/o LP) | **84.28** | - |
| SSD(Vehicle w/o LP+Dis LP) | 83.61 | **85.36** |
| TB(Vehicle w/o LP+Dis LP) | 84.23 | 84.11 |

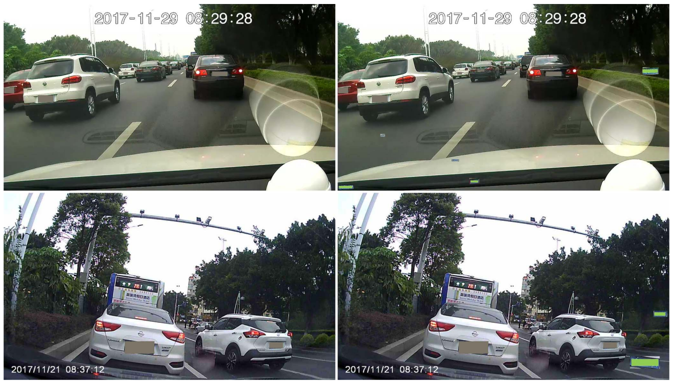

Fig. 6. Two Examples from VALID. The first column shows only removing the license plate by filling it with the mean of ImageNet. The second column shows further distributing the license plate randomly to other places. All recognizable license plates are manually blurred to protect the privacy.

TABLE III

DETECTION RESULTS (%) OF TRAINING VEHICLE AND LICENSE PLATE ON THE VALIDATION SET OF VALID WHEN SHARING CLASSIFIER AND REGRESSOR, ONLY SHARING REGRESSOR AS WELL AS SEPARATING BOTH CLASSIFIER AND REGRESSOR

| Method | $AP_{0.5}$ of Vehicle | $AP_{0.5}$ of LP |
|---|---|---|
| C_shared/R_shared | 87.47 | 73.43 |
| C_separated/R_shared | 85.31 | 84.86 |
| C_separated/R_separated | **87.93** | **87.41** |

757 images are randomly divided into training-validation and test set by 7:3.

### B. Observed Problem

We find it difficult to detect the vehicle and the license plate simultaneously using SSD [11], where license plate detection is largely affected by the vehicle. To evaluate the performance of the detection results, we adopt the general AP (Average Precision) as the evaluation protocol. To be specific, we follow the 11-point computation of the VOC2007 [36], where the detected bounding box is considered as correct if the IoU with the ground truth is more than 0.5.

Table I demonstrates the detection results on the test set of VALID with SSD300 [11] when training vehicle and license plate separately in two independent networks or simultaneously in one network. From the table, we can see that the AP of the license plate drops more than 14% when training vehicle and license plate together. Meanwhile, the AP of the vehicle is not affected. More experiments about the observed problem are described in the supplementary materials.

To illustrate the vehicle's effects on the license plate, all license plates are randomly distributed to other places to decouple the relationship between the vehicle and the license plate, making sure the distributed license plates are not overlapped with vehicles and each other. The license plates in the vehicles are replaced with the mean of ImageNet [31], as shown in Figure 6. From the first two rows of Table II, we can see that removing the license plate does not affect the vehicle. However, the performance of the license plate improves a lot after separating it from the vehicle, almost having the same performance as training license plate alone in Table I.

### C. Experiments With Detection Branch

As shown in Figure 1, with SSD [11], the confidence of the license plate is at a low level due to the inclusion relation of the vehicle and the license plate. When training vehicle and license plate with shared head layers, the confidence of each object is calculated as (4). Due to the inclusion relation, the classifier tends to classify candidate anchor boxes as vehicles. Furthermore, with SSD [11], the regressor is also class-agnostic, which can make the regression parameters unstable because they are influenced by the vehicle and the license plate simultaneously. Based on this, we first separate the classifier for the vehicle and the license plate as (5) and (6), and then also separate the regressor for two objects into two independent detection branches. Table III demonstrates the detection results of sharing classifier and regressor, only sharing regressor as well as separating both classifier and regressor, where C_shared/C_separated means sharing/separating classifier, and R_shared/R_separated means sharing/separating regressor. It can be seen that the performance of the license plate improves a lot after separating the classifier, which proves that the vehicle can affect license plate detection. Besides, the performance of the vehicle drops a bit after separating the classifier because we intuitively think the license plate can influence more on the regressor. Moreover, the performance of the vehicle and the license plate obtains further improvement after separating the regressor because of independent detection branches.

In consideration of the license plate's small size and fairly obvious features as well as the vehicle's various size and rather complex features, the license plates are detected with low-level features and the vehicles are localized with multi-level features, as shown in Figure 2. More details are described in the supplementary materials.

From the last row of Table II, our method achieves comparative performance with SSD [11] on the processed test

TABLE IV

ABLATION STUDY (%) ON THE TEST SET OF VALID

| Method(Dataset) | Classifier Separation | Two Branches | Anchor Clustering | Attention | Feature Fusion | $AP_{0.5}$ of Vehicle | $AP_{0.5}$ of LP |
|---|---|---|---|---|---|---|---|
| SSD(Vehicle) | | | | | | 83.96 | - |
| SSD(LP) | | | | | | - | 85.9 |
| SSD(Vehicle+LP) | | | | | | 83.77 | 71.86 |
| Ours(Vehicle+LP) | √ | | | | | 82.6 | 82.35 |
| | √ | √ | | | | 83.78 | 85.48 |
| | √ | √ | √ | | | 85.69 | 86.21 |
| | √ | √ | √ | √ | | 86.37 | 87.69 |
| | √ | √ | √ | √ | √ | **86.84** | **88.28** |

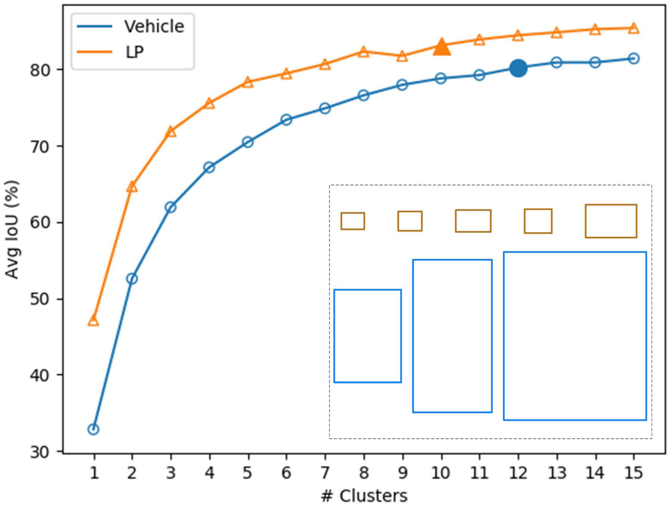

Fig. 7. Anchor clustering. We select 12 cluster centroids for the vehicle and 10 cluster centroids for the license plate. There are several representative cluster centroids of the vehicle and the license plate in the bottom-right corner.

set of VALID. Furthermore, from Table IV, we can see that detecting vehicle and license plate with two branches largely improves the detection performance of the license plate, almost achieving the performance of training license plate alone.

### D. Experiments With Anchor Design Strategy

We apply K-means clustering on the training set of VALID to generate anchor priors of the vehicle and the license plate respectively. With SSD [11], altogether 30 kind of anchor boxes are adopted, denoted as $A_{ssd} = \{4, 6, 6, 6, 4, 4\}$ for 6 head layers respectively. As demonstrated in Figure 7, we select 12 cluster centroids for the vehicle and 10 cluster centroids for the license plate, denoted as $A_{Vehicle} = \{4, 6, 1, 1, 1, 1\}$ and $A_{LP} = \{4, 6\}$ respectively. For each branch, we assign a close number of anchor boxes with SSD [11]. For the vehicle detection branch, the last 4 head layers share 2 cluster centroids. From Table V, the average IoU and spatial IoU of the license plate are all at a low level using the anchor design strategy of SSD. Our anchor clustering method achieves higher average IoU [16] and spatial IoU with fewer cluster centroids and anchor numbers, especially for the license plate. Furthermore, due to better matching with the ground truths, our strategy makes the network converge faster. More details of our anchor design strategy are described in the supplementary materials.

### E. Experiments With Attention and Feature Fusion

The spatial attention mechanism can highlight foreground information for better detection. As shown in Figure 3, the attention supervision is simply obtained by filling the ground truths. The predicted attention mask (Figure 4) is added before the classification and regression module, where the attention maps are fed into exponential operation and then have dot product with the feature maps. In this way, the regions of the vehicle and license plate are enhanced while the background is kept. Considering that deeper layers have larger receptive fields, attention on these layers may bring in extra noises. Only the bottom two layers of each branch are employed with attention. Furthermore, we also simply apply the feature fusion strategy (Figure 5) between the bottom two layers of each branch, as shown in Figure 2. More details are described in the supplementary materials.

Both the attention mechanism and feature-fusion strategy improve the detection performance for both two objects, as demonstrated in Table IV. However, the detection of the vehicle and license plate in real scenes is still unsolved due to size changes, pose variations and viewpoint changes, where scale-aware [37], graph matching [38]–[40] and multi-directional detection [7] methods can be consulted in the future.

### F. Comparative Experiments

For VALID, DETROIT and DOC, we compare Faster R-CNN[3] [13], YOLO [12], YOLOv2 [16], YOLOv3[4] [17] and SSD[5] [11] with our proposed method. The backbone of Faster R-CNN and SSD is set to VGG-16 [28], while the backbone of YOLO(v1-v3) remains unchanged. For our method, the experiment settings follow the settings of VALID, including the anchor clustering centroids.

As shown in Table VI, whether training vehicle and license plate separately or together, our method obtains the best performance for both three datasets. Moreover, with our method,

[3] https://github.com/jwyang/faster-rcnn.pytorch
[4] https://github.com/pjreddie/darknet
[5] https://github.com/amdegroot/ssd.pytorch

TABLE V

Average IoU (%) and Spatial IoU (%) ($N_A$ Means Anchor Numbers.)

| Box Generation | # Clusters of Vehicle/$N_A$ | # Clusters of LP/$N_A$ | Avg IoU of Vehicle | Avg IoU of LP | Spt IoU of Vehicle | Spt IoU of LP |
|---|---|---|---|---|---|---|
| SSD300 | 30/8732 | 30/8732 | 61.47 | 17.65 | 57.59 | 17.34 |
| Ours300 | **12/8077** | **10/7942** | **80.24** | **83.14** | **64.71** | **46.21** |

TABLE VI

Detection Results ($AP_{0.5}$, %) on the Test Set of VALID, DETROIT and DOC

| Training Dataset | Method | VALID | | DETROIT | | DOC | |
|---|---|---|---|---|---|---|---|
| | | Vehicle | LP | Vehicle | LP | Vehicle | LP |
| Vehicle | Faster R-CNN 300 [13] | 71.15 | - | 68.33 | - | **100** | - |
| | YOLO 320 [12] | 69.25 | - | 63.11 | - | 57.32 | - |
| | YOLOv2 320 [16] | 76.73 | - | 71.89 | - | 96.88 | - |
| | Fast YOLOv2 448 [16] | 75.02 | - | 65.11 | - | 96.88 | - |
| | Fast YOLOv3 320 [17] | 79.42 | - | 69.06 | - | **100** | - |
| | SSD 300 [11] | 83.96 | - | 71.37 | - | **100** | - |
| | Ours 300 | **86.13** | - | **71.92** | - | **100** | - |
| LP | Faster R-CNN 300 [13] | - | 54.95 | - | 62.25 | - | 59.56 |
| | YOLO 320 [12] | - | 66.75 | - | 64.82 | - | 49.18 |
| | YOLOv2 320 [16] | - | 80.75 | - | 73.95 | - | 96.68 |
| | Fast YOLOv2 448 [16] | - | 79.13 | - | 67.1 | - | 89.93 |
| | Fast YOLOv3 320 [17] | - | 79.69 | - | 69.30 | - | 90.62 |
| | SSD 300 [11] | - | 85.9 | - | 75.96 | - | 96.78 |
| | Ours 300 | - | **88.73** | - | **79.73** | - | **97.07** |
| Vehicle+LP | Faster R-CNN 300 [13] | 72.09 | 32.02 | 67.03 | 37.59 | ***100*** | 9.63 |
| | YOLO 320 [12] | 70.42 | 61.07 | 63.72 | 62.73 | 56.54 | 31.53 |
| | YOLOv2 320 [16] | 76.35 | 68.31 | 70.01 | 68.8 | 96.88 | 93.65 |
| | Fast YOLOv2 448 [16] | 74.32 | 60.15 | 64.51 | 62.75 | ***100*** | 82.51 |
| | Fast YOLOv3 320 [17] | 79.77 | 79.09 | 68.71 | 69.20 | ***100*** | 90.62 |
| | SSD 300 [11] | 83.77 | 71.86 | 71.51 | 71.52 | ***100*** | 95.46 |
| | Ours 300 | ***86.84*** | ***88.28*** | ***72.49*** | ***79.29*** | ***100*** | ***96.88*** |

the performance of the license plate improves greatly for both three datasets.

For other methods, except YOLOv3, they all have similar phenomena with SSD300 when training vehicle and license plate together. License plate detection is largely affected by the vehicle, while the vehicle is less affected. YOLOv3 divides the input images into many grids and each grid is responsible for detecting the object, so license plate detection is almost unaffected. For Faster R-CNN, the performance of the license plate drops dramatically for both three datasets, because Faster R-CNN is a two-stage network and license plate detection can be affected by the vehicle in both two stages.

Note: we do not experiment with large input images and more powerful backbone networks, because we hope to reduce the inference time and develop a real-time system for real applications.

### G. Additional Experiments

To evaluate the effectiveness of each detection branch, two independent networks for vehicle detection and license plate detection are trained separately, where one network only has the vehicle detection branch and another one only has the license plate detection branch. Two networks are trained with Udacity Dataset 1 and AOLP-LE respectively. The experiment settings are the same as VALID, including the anchor clustering centroids. Table VII proves our method of high accuracy and good generalization capability.

TABLE VII

Detection Results ($AP_{0.5}$, %) on the Test Set of Udacity Dataset 1 and AOLP-LE

| Method(Dataset) | Vehicle in Udacity Dataset 1 | LP in AOLP-LE |
|---|---|---|
| SSD300(Vehicle) | 69.03 | - |
| SSD300(LP) | - | 84.45 |
| Ours300(Vehicle) | **77.42** | - |
| Ours300(LP) | - | **87.07** |

### H. Analysis

We further evaluate whether the performance gains come from the better anchor design strategy. All 22 kind of anchors are applied to SSD300 [11] in a dense way, denoted as $A_{ssd(dense)} = \{8, 12, 1, 1, 1, 1\}$, which combines $A_{Vehicle}$ and $A_{LP}$ together. For a fair comparison, our method only conducts the two branches and anchor clustering strategies,

TABLE VIII

DETECTION RESULTS ($AP_{0.5}$, %) ON THE TEST SET OF VALID, DETROIT AND DOC (DENSE MEANS DENSE ANCHOR BOXES. OUR METHOD ONLY CONDUCTS TWO BRANCHES (TB) AND ANCHOR CLUSTERING (AC), WITHOUT ATTENTION AND FEATURE FUSION.)

| Method | Vehicle in VALID | LP in VALID | Vehicle in DETROIT | LP in DETROIT | Vehicle in DOC | LP in DOC |
|---|---|---|---|---|---|---|
| SSD300 | 83.77 | 71.86 | 71.51 | 71.52 | **100** | 95.46 |
| SSD300(Dense) | 85.51 | **87.04** | 68.69 | 73.32 | **100** | 95.96 |
| Ours300(TB+AC) | **85.69** | 86.21 | **72.18** | **78.74** | **100** | **96.78** |

TABLE IX

INFERENCE TIME, PARAMETERS AND FLOPS

| Method | Inference Time(ms) | Parameters(M) | FLOPS(G) |
|---|---|---|---|
| SSD300 | 23.02 | 22.77 | 31.96 |
| Ours300 | **21.98** | 23.33 | 32.45 |

without the attention and feature fusion modules. Table VIII demonstrates that the detection performance improves a lot with dense anchors, especially for the license plate. However, the performance of the vehicle in DETROIT declines significantly, because the anchor centroids obtained from VALID are over-fitting for DETROIT. With our method, it achieves better performance for all three datasets, which proves our method of good generalization capability.

### I. Inference Time

From Table IX, we can see that only about 0.56M parameters and 0.49G FLOPS are increased with our method, mainly from the feature-fusion module and the head network of two branches. However, our method takes less inference time, because two detection branches are running in parallel. Our head networks take less MAC(Multiplication and Addition), with 0.303G FLOPS for the vehicle branch and 0.299G FLOPS for the license plate branch, while SSD takes 0.361G FLOPS for both the vehicle and the license plate. Moreover, two detection branches have only one foreground class and it takes less NMS time, compared with the original SSD which has two foreground classes.

## V. CONCLUSION

In this paper, we are targeting to solve the problem that the vehicle affects license plate detection when detecting the vehicle and license plate simultaneously. We propose to separate the detection head networks into two independent branches, which improves the performance of the license plate dramatically. By adding a task-specific anchor design strategy, the network can obtain better predictions. Moreover, the attention mechanism and feature fusion strategy further enhance the detection performance. Finally, we validate our method of high accuracy, generalization capability and efficiency using images collected from real scenes and public datasets. For future work, we hope to evaluate whether our proposed multi-branch strategy can be applied to other prestigious frameworks, like YOLO and Faster R-CNN.

## REFERENCES

[1] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, May 2006.

[2] 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.

[3] C.-C. Tsai, C.-K. Tseng, H.-C. Tang, and J.-I. Guo, "Vehicle detection and classification based on deep neural network for intelligent transportation applications," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Honolulu, HI, USA, Nov. 2018, pp. 1605–1608.

[4] K.-H. Chen, T. D. Shou, J. K.-H. Li, and C.-M. Tsai, "Vehicles detection on expressway via deep learning: Single shot multibox object detector," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Chengdu, China, Jul. 2018, pp. 467–473.

[5] G.-S. Hsu, A. Ambikapathi, S.-L. Chung, and C.-P. Su, "Robust license plate detection in the wild," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Lecce, Italy, Aug./Sep. 2017, pp. 1–6.

[6] F. D. Kurpiel, R. Minetto, and B. T. Nassu, "Convolutional neural networks for license plate detection in images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 3395–3399.

[7] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 507–517, Feb. 2018.

[8] S. Kim, H. Jeon, and H. Koo, "Deep-learning-based license plate detection method using vehicle region extraction," *Electron. Lett.*, vol. 53, no. 15, pp. 1034–1036, 2017.

[9] Q. Fu, Y. Shen, and Z. Guo, "License plate detection using deep cascaded convolutional neural networks in complex scenes," in *Proc. Int. Conf. Neural Inf. Process.* in Lecture Notes in Computer Science, vol. 10635, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. El-Alfy, Eds. Guangzhou, China: Springer, 2017, pp. 696–706.

[10] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proc. Eur. Conf. Comput. Vis.* in Lecture Notes in Computer Science, vol. 11216, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Munich, Germany: Springer, 2018, pp. 593–609.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 9905. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[12] 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.*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 8689, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Zurich, Switzerland: Springer, 2014, pp. 818–833.

[15] T.-Y. Lin and S. Maji, "Visualizing and understanding deep texture representations," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2791–2799.

[16] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6517–6525.

[17] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[18] J. Wang, Y. Yuan, and G. Yu, "Face attention network: An effective face detector for the occluded faces," 2017, *arXiv:1711.07246*. [Online]. Available: https://arxiv.org/abs/1711.07246

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[21] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[22] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Sep. 2015, pp. 1440–1448.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[24] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with LLE-CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 426–434.

[25] S. Han, Y. Han, and H. Hahn, "Vehicle detection method using Haar-like feature on real time system," *Int. J. Electr. Comput. Eng.*, vol. 3, no. 11, pp. 1957–1961, Nov. 2009.

[26] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, Sep. 2012.

[27] 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.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, B. Yoshua and Y. Lecun, Eds. San Diego, CA, USA: OpenReview.net, 2015.

[29] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2874–2883.

[30] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, *arXiv:1712.00960*. [Online]. Available: https://arxiv.org/abs/1712.00960

[31] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, B. Yoshua and Y. Lecun, Eds. San Diego, CA, USA: OpenReview.net, 2015.

[33] A. Kuznetsova *et al.*, "The open images dataset V4: Unified image classification, object detection, and visual relationship detection at scale," 2018, *arXiv:1811.00982*. [Online]. Available: https://arxiv.org/abs/1811.00982

[34] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Jun. 2013, pp. 554–561.

[35] 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.

[36] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[37] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," 2019, *arXiv:1901.01892*. [Online]. Available: https://arxiv.org/abs/1901.01892

[38] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.

[39] J. Yan, C. Li, Y. Li, and G. Cao, "Adaptive discrete hypergraph matching," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 765–779, Feb. 2018.

[40] S. Ge, S. Zhao, C. Li, and J. Li, "Low-Resolution face recognition in the wild via selective knowledge distillation," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2051–2062, Apr. 2019.

**Song-Lu Chen** received the B.Sc. and M.Sc. degrees in computer science from the University of Science and Technology Beijing, China, in 2014 and 2017, respectively, where he currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include pattern recognition and object detection.

**Chun Yang** received the B.Sc. and Ph.D. degrees in computer science from the University of Science and Technology Beijing, China, in 2011 and 2018, respectively. He is currently a Faculty Member with the School of Computer and Communication Engineering, University of Science and Technology Beijing. His current research interests include pattern recognition, classifier ensemble, and document analysis and recognition.

**Jia-Wei Ma** received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2018, where he is currently pursuing the master's degree with the Department of Computer Science and Technology. His research interests include pattern recognition and object detection.

**Feng Chen** received the B.Sc. degree in automation from the University of Science and Technology Beijing, China, in 1999. He set up Actions Semiconductor as a Founding Engineer in 2001. As the Director, he led the team to design the Game chipset of ultra large scale integration, which has successfully entered the Japanese market. He set up All Winner Technology as a Co-Founder in 2007. As the Director, he led the team to design graphics and display processors of low bandwidth and low power with in-memory design ideas and gained the leading position in the industry. He is currently a Co-Founder of EEasy Technology Company, Ltd. His research interests include image and graphic, and display processor.

**Xu-Cheng Yin** (M'10–SM'16) received the B.Sc. and M.Sc. degrees in computer science from the University of Science and Technology Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2006.

He was a Visiting Researcher with the School of Computer Science, University of Massachusetts Amherst, USA, from January 2013 to January 2014 and from July 2014 to August 2014, respectively. He is currently a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing. He has published more than 50 research papers (IEEE TPAMI, IEEE TIP, *Information Sciences*, IJCAI, SIGIR, MM, CIKM, ICDAR and ICPR). His research interests include pattern recognition, computer vision, and document analysis and recognition. His team won the first place of both Text Localization in Real Scenes and Text Localization in Born-Digital Images in the ICDAR 2013 Robust Reading Competition, the First Place of both End-To-End Text Recognition in Real Scenes (Generic) and End-To-End Text Recognition in Born-Digital Images (Generic) in the ICDAR 2015 Robust Reading Competition, and the First Place of ICDAR 2017 Robust Reading Competition Challenge on COCO-Text (End-To-End Text Recognition from Real Scenes).