

## Article

# License Plate Detection with Attention-Guided Dual Feature Pyramid Networks in Complex Environments

Yu-Jie Xiong \*, Yong-Bin Gao \*, Jun-Qing Zhang and Jian-Xin Ren

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

\* Correspondence: xiong@sues.edu.cn (Y.-J.X.); gaoyongbin@sues.edu.cn (Y.-B.G.)

**Abstract:** License plate detection plays a significant role in intelligent transportation systems. Convolutional neural networks have shown a remarkable performance and made significant progress for the detection task. Despite these outstanding achievements, license plate detection in complex environments is still a challenging task, due to the noisy background, unpredictable environments and varying shapes and sizes of the license plates. In order to improve the performance of license plate detection in complex environments, we propose a novel approach using an attention-guided dual feature pyramid and a cascaded positioning strategy. At first, the original features of images are extracted by the residual network. In order to make sure that each feature map contains higher- and lower-level semantic information, we utilize a bottom-up and a top-down pathway, respectively. Meanwhile, the proposed attention-guided dual feature pyramid network is used to receive the extracted features for a multilevel feature fusion. Our proposed attention-guided modules contain both spatial and channel attention. Attention-guided modules deduce the attention weights according to channel and spatial dimensions and multiply the calculated result with the input to obtain the refined feature maps. Then, a region proposal network is used to generate the candidate regions for the license plates. Finally, a cascaded positioning network is utilized to obtain the final locations of the license plates. To validate the effectiveness of the proposed method, we conducted a series of experiments on different public datasets. Experiments on AOLP and CCPD validated the effectiveness of our proposed method.

**Keywords:** convolutional neural network; license plate detection; cascaded training; attention-guided



**Citation:** Xiong, Y.-J.; Gao, Y.-B.; Zhang, J.-Q.; Ren, J.-X. License Plate Detection with Attention-Guided Dual Feature Pyramid Networks in Complex Environments. *Electronics* **2022**, *11*, 3895. <https://doi.org/10.3390/electronics11233895>

Academic Editor: George A. Papakostas

Received: 15 August 2022

Accepted: 13 November 2022

Published: 25 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the rapid development of national economies, the standard of living has been improving continuously in recent decades. However, more and more vehicles have also brought about a series of problems such as vehicle violations, traffic accidents and parking difficulties. In order to build an intelligent traffic management system, modern intelligent transport systems (ITSs) have been developed, and automatic license plate recognition (ALPR) [1] has been widely employed. ALPR searches the position of a license plate automatically from the collected images and extracts the relevant information of the license plate. Although license plate recognition is capable of simple restricted application scenarios, license plate images taken in complex environments are still very challenging. Research on license plate detection in complex application scenarios can improve ITSs to reduce traffic pressure and safety risks.

Traditional license plate detection methods are often based on handcrafted image features. Texture-based methods can also be used to detect license plates as plate regions generally have unconventional pixel texture distribution. Texture-based methods often adopt more discriminative characteristics than edge-based methods and color-based methods, but they have a high computing complexity. These handcrafted image features have achieved good results in specific application scenarios but are mostly dependent on the

experience of feature designers. Furthermore, these traditional methods do not have a good adaptability in complex environments. Fortunately, semantic segmentation techniques with deep learning provide us a new way to achieve the goal of license plate detection.

In order to solve the problem of license plate detection in complex environments, this paper proposes a license plate detection approach, which combines a dual pyramid feature fusion and cascade positioning. Firstly, we extract the primary features of the input image by using a residual network. The extracted primary features are sent to the dual pyramid feature networks for a multilayer feature fusion. Then, a region proposal network based on the prior shape is used to generate the regions of interest. Finally, the generated regions of interest are sent to the cascade positioning network to obtain accurate license plate detection results. In order to verify the effectiveness of the license plate detection approach proposed in this paper, a large number of experiments are carried out on the AOLP dataset and the CCPD dataset.

## 2. Related Work

### 2.1. License Plate Detection

The license plate detection aims at localizing a license plate in an image in the form of a bounding box. It can be broadly classified into traditional approaches and deep learning approaches. Existing traditional approaches for license plate detection are mostly dependent on handcrafted features and can be roughly categorized into four groups: texture-based approaches, color-based approaches, character-based approaches and edge-based approaches [2]. In [3], a sliding concentric window (SCW) was proposed to address the problem of license plate detection. The authors also used operator context scanning (OCS) to reduce the detection time. Color features can also be designed to detect license plates. These approaches generally localize license plates by detecting the difference in color between the license plate and the body of vehicles. Ashrari et al. [4] used a color-based method for Iranian license plate detection based on a color–geometric template. As license plates include a string of characters, many researchers also utilized character-based approaches to detect license plates. In [5], maximally stable extremal regions (MSER) were adopted to first extract the character features. Then, a conditional random field (CRF) was used to represent the relationship among candidate characters. Considering the fact that a license plate is generally in a rectangular shape with a special ratio, edge-based approaches are widely used for license plate detection. In [6], Yuan et al. proposed a robust line density filter algorithm to extract the edge features of the candidate regions. After that, a linear support vector machine (SVM) was utilized to verify these regions.

In the last few years, many convolutional neural network (CNN)-based frameworks such as YOLO [7], Faster R-CNN [8], SSD [9] and FCN [10] have been used for license plate detection. Based on these frameworks, many related research studies on license plate detection have been proposed. Researchers have employed CNNs for feature extraction and classification at the same time. In [11], a YOLO-based framework was proposed for a multidirectional license plate detection. In [12], Xiang et al. used an FCN-based model for localizing Chinese license plates. In [13], Kim improved Faster R-CNN and developed a license plate detection method based on vehicle features. In order to obtain a higher detection performance, researchers have integrated some effective structures [14–16] into the original framework and achieved great success.

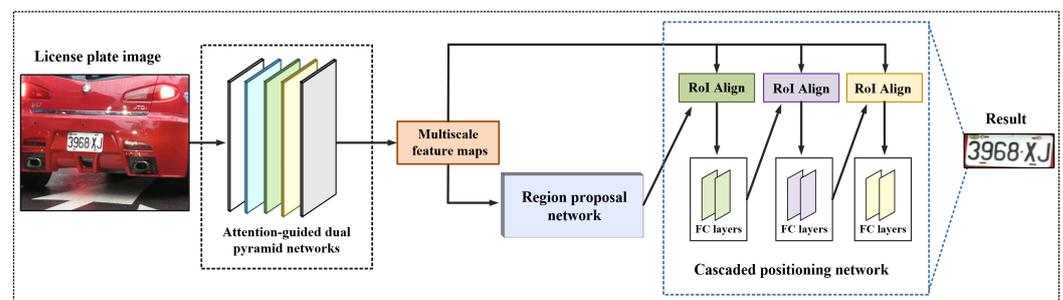
### 2.2. License Plate Recognition

License plate recognition aims at recognizing the license plate characters after license plate detection. Existing license plate recognition approaches can broadly be divided into segmentation-based approaches and unsegmented-based approaches. In segmentation-based approaches [17], character segmentation is the link between license plate detection and recognition and plays an import role in the final recognition results. Thus, these approaches tend to segment the license plate regions to individual characters first and then recognize them. However, the segmentation performance are generally influenced

by complex environments. Interference factors such as weak lighting and blurring lead to poor segmentation quality, which further affect recognition accuracy. In recent years, unsegmented-based approaches [18,19] have become very popular in license plate recognition. In [20], the license plate characters were treated as a sequence. The authors employed a convolutional neural network (CNN) and a bidirectional recurrent neural network (BRNN) to encode sequential features and then decoded them by connectionist temporal classification (CTC). The authors also adopted a similar nonsegmentation algorithm, but BRNN was replaced by another CNN when extracting the sequential features [21].

### 3. Methodology

The framework of our proposed method is based on Mask R-CNN. Mask R-CNN employs ResNet [15] and FPN [16] as the backbone to extract features. Similarly, as illustrated in Figure 1, our proposed method consists of three main parts: attention-guided dual feature pyramid networks (ADFPN) for the feature extraction, a region proposal network (RPN) for candidate regions generation and a cascaded positioning network (CPN) for final license plate detection.



**Figure 1.** Illustration of the proposed method.

The main contributions of our method are as follows:

- We propose attention-guided dual feature pyramid networks for multilevel feature extraction and fusion.
- We propose a cascaded positioning strategy for the coarse-to-fine license plate positioning.
- Experiments on real-world datasets validate the rationality and effectiveness of our proposed approach.

#### 3.1. Attention-Guided Dual Feature Pyramid Networks

Currently, ResNet is used to extract low-level features first and then an FPN is utilized to fuse multilevel feature maps. However, the original FPN can only use the information from the current and higher-level feature maps. Yet, the information from the lower feature maps cannot be included. Thus, it may fail to make full use of the feature information of each scale. The detailed illustration of our proposed ADFPN is shown in Figure 2.

The ADFPN consisted of two parts: the original ResNet-101-FPN backbone and the additional pathways. ResNet-101 is composed of five feature maps from high to low resolution, namely,  $\{C_1, C_2, C_3, C_4, C_5\}$ , at five scales. The features were extracted by a bottom-top pathway and  $\{C_2, C_3, C_4, C_5\}$  were sent to the original FPN as the outputs of ResNet-101. Each feature map was reduced to the same spatial size via a  $1 \times 1$  convolution. In the up-down pathway of the FPN, feature maps  $\{C_2, C_3, C_4, C_5\}$  were merged into  $\{P_2, P_3, P_4, P_5\}$  through the lateral connections. The entire process was as follows.

$$P_i = \begin{cases} Conv_{1 \times 1}(C_i), i = 5 \\ Upsample(P_{i+1}) + Conv_{1 \times 1}(C_i), i = 2, 3, 4 \end{cases} \quad (1)$$

where  $Conv_{1 \times 1}$  is a  $1 \times 1$  convolution and  $Upsample$  is a bilinear upsampling operation.

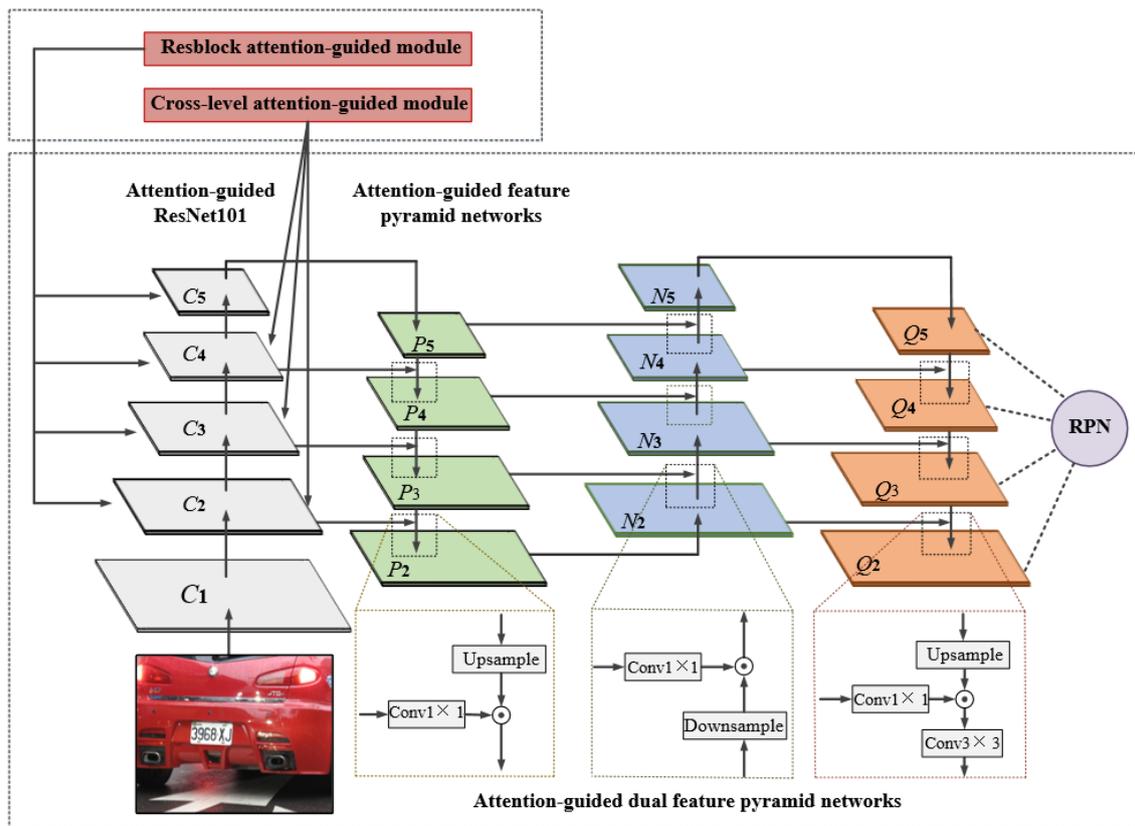


Figure 2. Illustration of attention-guided dual feature pyramid networks.

Compared with the original ResNet-FPN backbone, in order to make sure that each feature map contains the semantic information of its higher- and lower-level feature maps, we utilized a bottom-up and a top-down pathway, respectively. In the additional bottom-up pathway, the feature maps  $\{P_2, P_3, P_4, P_5\}$  from the original FPN were merged into  $\{N_2, N_3, N_4, N_5\}$  through the lateral connections and downsampling operation. Similarly, in the top-down pathway, the feature maps  $\{N_2, N_3, N_4, N_5\}$  from the bottom-up pathway were also merged into  $\{Q_2, Q_3, Q_4, Q_5\}$  through the upsampling operation and lateral connections. Finally, we used the  $3 \times 3$  convolution operation to merge the feature maps and produce the final feature maps. The entire process was as follows.

$$N_i = \begin{cases} Conv_{1 \times 1}(P_i), i = 2 \\ Downsample(N_{i+1}) + Conv_{1 \times 1}(P_i), i = 3, 4, 5 \end{cases} \tag{2}$$

$$Q_i = \begin{cases} Conv_{1 \times 1}(N_i), i = 5 \\ Upsample(Q_{i+1}) + Conv_{1 \times 1}(N_i), i = 2, 3, 4 \end{cases} \tag{3}$$

where  $Conv_{1 \times 1}$  is a  $1 \times 1$  convolution,  $Conv_{3 \times 3}$  is a  $3 \times 3$  convolution,  $Upsample$  is a bilinear upsampling operation,  $Downsample$  is a  $2 \times 2$  max-pooling operation and  $+$  is an elementwise addition.

### 3.2. Attention-Guided Modules

As mentioned in [22], in the feature extraction stage, low-level features generally have uncertain semantic information, which may affect the accuracy of license plate detection. Although the proposed dual feature pyramid networks can better express the semantic information of the high- and low-level feature maps by adding new pathways, as a result, it also leads to redundant semantic information in the feature fusion. To solve the problem, we introduced attention-guided modules into the dual feature pyramid networks. As we know, attention plays an important role in human perception. At present, the attention

mechanism is widely used in mainstream object detection [23]. It focuses on important local information and inhibits the expression of features that have few effects on license plate detection. Hence, the usage of an attention mechanism can enhance the feature representation of networks and improve the robustness of the detection.

Our proposed attention-guided modules contained both spatial and channel attention. For an input feature map, the attention-guided modules deduced the attention weight of the feature map according to the channel and spatial dimensions first and then multiplied the calculated result with the input feature map to obtain the refined feature map. The attention-guided modules were defined as:

$$K' = Channel(K) \otimes K \tag{4}$$

$$Attention(K') = Spatial(K') \otimes K' \tag{5}$$

$$Channel(K) = f(MLP(Avg(K)) + MLP(Max(K))) \tag{6}$$

$$Spatial(K') = f(Conv_{7 \times 7}(Avg(K') \oplus Max(K'))) \tag{7}$$

where  $K$  is the input feature map,  $K'$  is the feature map through the channel attention module,  $\otimes$  is a multiply operation,  $Attention(K')$  is the feature map through the attention-guided module,  $Avg$  is a global average pooling,  $Max$  is a global max-pooling,  $+$  is an elementwise addition,  $\oplus$  is a concatenation operation,  $f$  is a sigmoid function,  $Conv_{7 \times 7}$  is a  $7 \times 7$  convolution operation and  $MLP$  is a two-layer neural network.

According to the different positions, the attention-guided modules were divided into two categories: the Resblock attention-guided module and the cross-level attention-guided module. In order to improve the effectiveness of features in the primary feature extraction, we added the attention-guided module into ResNet-101 as the Resblock attention-guided module. ResNet has two basic residual blocks: the identity block and the convolutional block. The dimensions of the inputs and outputs of the feature maps in the identity block remain the same, while the convolutional block can change the dimension of the feature maps. The proposed Resblock attention-guided module in each block is illustrated in Figure 3.

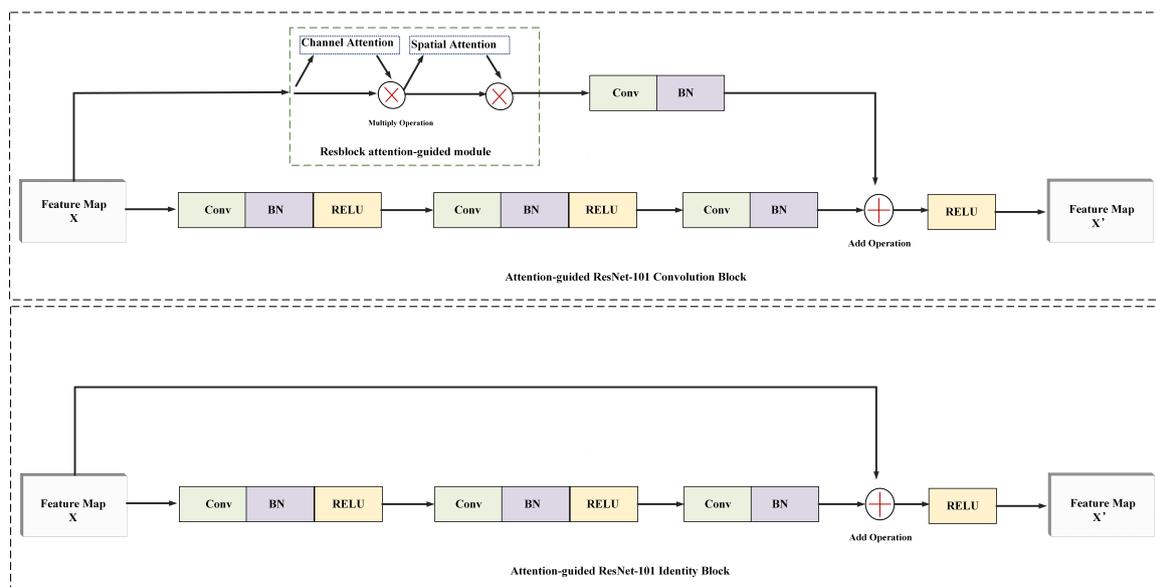
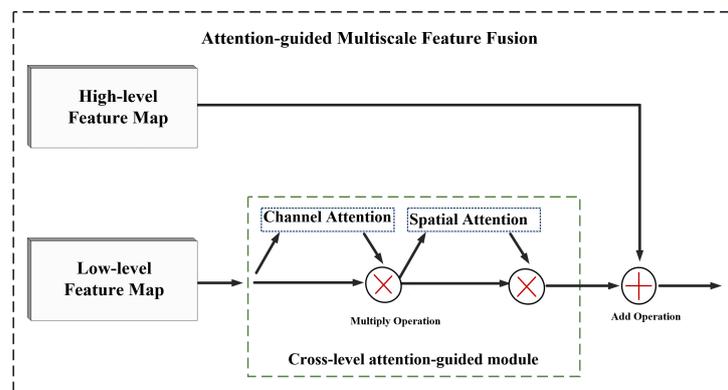


Figure 3. An example of Resblock attention-guided module.

As illustrated in Figure 4, we also added a cross-level attention-guided module into the FPN. The high-level feature map was combined with the low-level feature map by the cross-level attention-guided module. This module used high-level features to provide pixel-

level attention for low-level feature maps. Important information could be represented as much as possible with limited parameters.



**Figure 4.** An example of cross-level attention-guided module.

### 3.3. Candidate Regions Generation

Furthermore, the candidate proposals were generated by a region proposal network (RPN). Specifically, the output  $Q_i$  of the ADFPN was sent to the RPN for candidate regions generation. In the FPN, the scales of the anchors were set to  $\{32, 64, 128, 256, 512\}$ . The anchors of multiple aspect ratios were set to  $\{0.5, 1, 2\}$  at each stage.

### 3.4. Cascaded Positioning Network

The RoI Align layer was used to extract the localization of candidate regions. It intercepted the corresponding positions of candidate regions from feature maps according to their location information, and pooled them into unified feature maps of dimension  $7 \times 7$  through the bilinear interpolation algorithm. Then, the unified feature maps were sent to the full connection layer. In the full connection layer, the network completed the two branches of candidate box classification and regression through two full connection layers with 1024 channels. The whole location network used the extracted license plate features as input and outputted the location information of the license plate.

In order to obtain high-quality detection results, a cascaded positioning mechanism was introduced into the original network. The new network consisted of three positioning networks. In the cascaded positioning network, each positioning network had the same structure with different Intersection over Union (IoU) thresholds. An Intersection over Union (IoU) threshold was used to define positive/negative samples. The specific calculation formula of the IoU is as follows.

$$\text{IoU} = \frac{R_{\text{pred}} \cap R_{\text{gt}}}{R_{\text{pred}} \cup R_{\text{gt}}} \quad (8)$$

where  $R_{\text{pred}}$  is the boundary box area detected by the training model and  $R_{\text{gt}}$  is the boundary box area of the data annotation truth value. When the IoU value was set to 0.5 in the experiment, for a license plate area, if the IoU between the boundary box detected by the model and the true boundary box was larger than 50%, it was judged to be a positive sample, otherwise it was a negative sample.

Three different IoU thresholds of 0.5, 0.6 and 0.7 were set up at each stage. The output of the location network at each stage was used as the input of the location network at the next stage to gradually improve the accuracy of the location network output after cascading. Based on the above ideas, different from the original network which only used the threshold value of 0.5 IoU as the defining standard for positive and negative samples, the new positioning network improved the accuracy of the regression frame each time. By adjusting the regression frame, a positive sample with higher IoU was found at the next stage, so as to improve the accuracy of the detection frame.

### 3.5. Training Objective

In the training stage, the proposed method used four loss functions to evaluate the loss between the predicted value and the ground truth. The total loss function was as follows.

$$L = L_{cls-r} + L_{bbx-r} + \sum_{i=1}^3 (L_{cls_i} + L_{bbx_i}) \quad (9)$$

where  $L_{cls-r}$  and  $L_{bbx-r}$  represent the loss of front/back scene classifications and coordinates of RPN, respectively, and  $L_{cls_i}$  and  $L_{bbx_i}$  represent the loss of classification and bounding box of the cascaded positioning network at each stage.

$$L_{cls-r} = -\frac{1}{N_{cls}} \sum_{i=1}^N \log[p_i p_i^* + (1 - p_i)(1 - p_i^*)] \quad (10)$$

$$L_{bbx-r} = \frac{1}{N_{reg}} \sum_{i=1}^{N_{reg}} p_i^* R(t_i, t_i^*) \quad (11)$$

where  $N_{cls}$  is the number of selected anchors,  $R$  is a smooth L1 loss function,  $p_i$  is the probability that anchors are predicted as true and  $p_i^*$  is the ground truth label. The RoI Align layer outputted the uniform size of the region of interest to the final classification and bounding box regression. The losses in this stage were as follows.

$$L_{cls_i} = -\frac{1}{N_{cls}} \sum_{k=1}^{N_{cls}} \log[p_k p_k^* + (1 - p_k)(1 - p_k^*)] \quad (12)$$

$$L_{bbx_i} = \frac{1}{N_{reg}} \sum_{k=1}^{N_{reg}} p_k^* R(t_k, t_k^*) \quad (13)$$

where  $N_{cls}$  is the number of predicted objects,  $N_{reg}$  is the number of the bounding boxes and  $p_k$  is the probability of positive samples predicted when training.

## 4. Experiments

In this section, we demonstrate several experiments that reflect the various aspects of the proposed method. Our proposed method was implemented with python 3.6.9 and TensorFlow 1.9.0. The details of the optimizer and hyperparameters for our proposed method are also provided in Table 1.

**Table 1.** The details of the optimizer and hyperparameters for our proposed framework.

Parameter	Setting
Optimizer	SGD Optimizer
Learning rate	0.00075
Momentum	0.9
Weight decay (L2 regularization)	0.0001
Maximum number of iterations	90,000
Confidence threshold	0.7

### 4.1. Datasets

Two different datasets were used to evaluate the performance of the proposed method. The datasets were composed of license plate images captured under various illumination conditions. AOLP [24] includes license plates with horizontal angles and different tilting angles, as well as license plates with complex urban traffic background. CCPD [19] is used for license plate detection in rainy, snowy and foggy weather, nonuniform lighting conditions and on fuzzy days.

AOLP [24] is a widely used open-source public license plate detection dataset. The dataset consists of 2049 samples of Taiwanese license plate images. The AOLP dataset is divided into three subdatasets: AC (Access Control) dataset, LE (Law Enforcement) dataset and RP (Road Patrol) dataset. The above datasets contain 681, 757 and 611 samples, respectively. Since most of the sample vehicles in the AC dataset were captured when they passed through a crossing at restricted driving speeds, the samples were all horizontal license plate images. The LE dataset samples are from vehicles traveling in urban traffic, including pedestrians, street lamps and road signs with complex road backgrounds. The RP dataset samples are mostly tilted license plate images. Samples of the AOLP dataset were randomly divided into a training dataset and a test dataset. In order to ensure the consistency of the distribution of the three subdatasets in the training dataset and the test dataset, the subdatasets were randomly divided according to the same proportion. The experimental training dataset consisted of 1740 license plate images, and the test dataset consisted of 309 license plate images.

The CCPD (Chinese City Parking Dataset) dataset is a public Chinese license plate dataset. This dataset is a new large-scale comprehensive and diversified domestic license plate dataset proposed by Xu et al. [19] at ECCV2018, with more than 250,000 samples. The sample data were collected from various parking lots in China, and all the image samples have a resolution of  $720 \times 1160$ . Images were taken at different times and in different lighting conditions (such as day, evening and night), different climate environments (such as rainy and snowy days) and other complex environments. In order to verify the performance of the algorithm under different illumination conditions, we selected two subdatasets of CCPD-Blur and CCPD-Weather in the CCPD license plate dataset to verify the performance and robustness of the proposed method in complex environments. The interference factors of jitter, blur and nonuniform illumination existed in the sample of CCPD-Blur dataset. The interference factors of extreme weather conditions existed in the sample of CCPD-Weather dataset.

Five characteristic license plate datasets were used in our experiments. Due to the obvious differences and challenges in the samples of different license plate datasets, the effectiveness of the proposed method under different illumination conditions in complex environments could be verified better.

#### 4.2. Evaluation Metrics

Because of the similarity between license plate object detection and general text detection application scenarios in natural scenes, the evaluation metrics in [18] were used to measure the performance of our proposed method in this paper. In total, four evaluation metrics were used to evaluate the detection performance of the method, includes precision, recall and F-measure.

Precision is defined as the ratio of the number of license plate samples predicted by the detection algorithm to the number of samples predicted by the algorithm. Recall is defined as the ratio of the number of license plate samples predicted by the detection algorithm to the number of samples in the total test dataset. The F-measure is an evaluation metric of comprehensive detection performance. The evaluation metrics are as follows:

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$F - measure = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (16)$$

where  $TP$  is the number of license plate samples detected as positive samples that are actually positive samples,  $FP$  is the number of license plate samples detected as positive

samples that are actually negative samples and  $FN$  is the number of license plate samples detected as negative samples that are actually positive samples.

#### 4.3. Effect of the Cascaded Positioning Network

Figure 5 shows the detection results of the cascaded positioning network in different stages. In the training stage, the original network calculated the IoU of the RoIs and then defined the positive and negative training samples according to a IoU threshold of 0.5. Compared with the single-stage training, the cascaded positioning network was divided into three stages when training the classifier and regression of the network. This kind of training mode can get high-quality candidate regions by increasing the IoU threshold for defining positive and negative samples.



**Figure 5.** Results of different stages of the cascaded positioning network.

According to the comparison of the detection results shown in Figure 5, the detection results of the third-stage positioning network were more precise than those of the first-stage positioning network and second-stage positioning network. In turn, the IoU values of the detection results in Figure 5a–c have an increasing tendency. Therefore, it indicates that the detection results can be improved in different stages. Overall, the cascaded positioning network improved the detection results stage by stage to obtain a high-quality detection.

#### 4.4. Ablation Study

The proposed method combined three different improvement strategies, including dual feature pyramid networks, the attention-guided module and the cascaded positioning network. To validate the effectiveness of these strategies proposed in this paper, we trained the model on the AOLP dataset and conducted four experiments. The results of the experiments using different improved strategies are shown in Table 2.

**Table 2.** Ablation study on AOLP.

Dual Feature Pyramid Networks	Attention-Guided Model	Cascade Positioning Network	Precision (%)	Recall (%)	F-Measure (%)
			96.86	99.67	98.25
✓			97.47	99.67	98.56
✓	✓		98.40	99.67	99.03
✓	✓	✓	100.00	99.67	99.83

According to the observation and comparison of the precision, recall and F-measure, it shows that the detection performance of the proposed method could be improved by all three strategies. The improvement of the precision was the most obvious. In experiments, all results obtain a high recall at the low IoU threshold. At first, we trained the original Mask R-CNN. The precision and F-measure achieved 96.86% and 98.25%, respectively. Then, the dual feature pyramid networks were added. The precision and F-measure achieved 97.47% and 98.56%, respectively. After that, the attention-guided dual feature pyramid networks were employed for the feature fusion. The model obtained better performance. At last, we employed the attention-guided dual feature pyramid networks and the cascaded positioning network. The model achieved 100% precision and the F-measure was up to 99.83%. Overall, the proposed method could further improve the performance of license plate detection and all strategies had a certain effectiveness.

#### 4.5. Comparison with Other Approaches

In this section, we compare our proposed method with other approaches on the AOLP dataset. Tables 3–5 show the comparison of the detection results of different algorithms on three AOLP subdatasets, AC, LE and RP, with an IoU threshold of 0.5, respectively. For the dataset division of this part of the experiment, this paper followed the principle in [20]: samples in LE and RP datasets were used as training in Table 3 to evaluate the detection performance of the algorithm on the AC dataset. Tables 4 and 5 were used in the same way. It can be seen from the comparison of the detection results in Tables 3–5 that the IoU threshold of 0.5 was taken as the unified evaluation standard. According to Table 3, the precision and recall of the proposed method for the AC dataset were 99.71% and 99.56%, and the F-measure of the proposed method was better than that of the other approaches.

**Table 3.** Detection results on AOLP-AC.

Method	Precision (%)	Recall (%)	F-Measure (%)
Hsu et al. [24]	91.00	96.00	93.43
Selmi et al. [25]	92.60	96.80	94.65
Li et al. [20]	98.53	98.38	98.45
Selmi et al. [26]	99.30	99.40	99.35
Ours	99.71	99.56	99.63

**Table 4.** Detection results on AOLP-LE.

Method	Precision (%)	Recall (%)	F-Measure (%)
Hsu et al. [24]	91.00	95.00	92.96
Selmi et al. [25]	93.50	93.30	93.39
Li et al. [20]	97.65	97.62	97.68
Ours	99.19	97.62	98.40

**Table 5.** Detection results on AOLP-RP.

Method	Precision (%)	Recall (%)	F-Measure (%)
Hsu et al. [24]	91.00	94.00	92.48
Selmi et al. [25]	92.90	96.20	94.52
Li et al. [20]	95.28	95.58	95.43
Selmi et al. [26]	98.90	98.80	98.85
Ours	99.84	99.51	99.67

As shown in Table 4, the precision, recall and F-measure of the proposed method were 99.19%, 97.62% and 98.40%, respectively. The precision was improved by 1.54% compared with the algorithm in [20], and the recall rate was the same as that in [20]. Experimental results showed that in order to ensure the accuracy of the detection results, the proposed method sacrificed some recall accuracy.

However, in the comparison of the detection results of the RP dataset in Table 5, the method presented in this paper had excellent performance in precision and recall rate. We also compare the proposed method with SSD [9,11], Faster R-CNN [13], Mask R-CNN [14,27] and YOLOv3 [28] on various datasets, and the results showed that the proposed method achieved satisfactory results.

Table 6 shows the results of the F-measure on AOLP, CCPD-Weather and CCPD-Blur. Compared with [9,11,14], the proposed method improved the detection performance to a certain extent on three different license plate datasets, especially on the AOLP dataset. Compared with [28], our method improved by 0.33%, 0.23% and 0.28% on AOLP, CCPD-Weather and CCPD-Blur, respectively. Experimental results showed that the proposed method was stable under different conditions and located license plates precisely.

**Table 6.** F-measure on AOLP, CCPD-Weather and CCPD-Blur.

Method	Dataset		
	AOLP	CCPD-Weather	CCPD-Blur
Xie et al. [11]	99.47	-	-
Molina et al. [27]	98.98	-	-
Faster R-CNN [13]	98.87	97.89	96.88
Mask R-CNN [14]	98.56	98.84	97.20
SSD [9]	98.72	98.94	97.91
YOLOv3 [28]	99.50	99.56	98.65
Ours	99.83	99.79	98.93

## 5. Conclusions

Experiments showed that the proposed method detected the vehicle license plate object and achieved a high-quality positioning by combining the ADFPN and the cascade positioning structure strategy, which made the detection more effective and accurate. In the future, we will try to use polygons to better fit the shape of the license plate object to achieve a better detection. At the same time, how to achieve high-precision license plate detection while ensuring an adequate calculation speed will be our future research direction.

**Author Contributions:** Methodology, Y.-J.X.; Validation, J.-X.R.; Investigation, Y.-J.X.; Resources, Y.-J.X.; Writing—original draft, J.-Q.Z.; Project administration, Y.-B.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** Supported by: National Key Research and Development Program of China (2020AAA0109300); National Natural Science Foundation of China (62006150); Science and Technology Commission of Shanghai Municipality (21DZ2203100); ECNU special project of cultural inheritance and innovation (0400-22301-512200).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Du, S.; Ibrahim, M.; Shehata, M.; Badawy, W. Automatic license plate recognition (ALPR): A State-of-the-Art review. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 311–325. [[CrossRef](#)]
2. Zhou, W.; Li, H.; Lu, Y.; Tian, Q. Principal visual word discovery for automatic license plate detection. *IEEE Trans. Image Process.* **2012**, *21*, 4269–4279. [[CrossRef](#)] [[PubMed](#)]
3. Giannoukos, I.; Anagnostopoulos, C.N.; Loumos, V.; Kayafas, E. Operator context scanning to support high segmentation rates for real time license plate recognition. *Pattern Recognit.* **2010**, *43*, 3866–3878. [[CrossRef](#)]
4. Ashtari, A.H.; Nordin, M.J.; Fathy, M. An Iranian license plate recognition system based on color features. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1690–1705. [[CrossRef](#)]
5. Li, B.; Tian, B.; Li, Y.; Wen, D. Component-based license plate detection using conditional random field model. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1690–1699. [[CrossRef](#)]
6. Yuan, Y.; Zou, W.; Zhao, Y.; Wang, X.; Hu, X.; Komodakis, N. A robust and efficient approach to license plate detection. *IEEE Trans. Image Process.* **2017**, *26*, 1102–1114. [[CrossRef](#)] [[PubMed](#)]
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the 2016 European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
10. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
11. Xie, L.L.; Ahmad, T.; Jin, L.; Liu, Y.; Zhang, S. A new CNN-based method for multi-directional car license plate detection. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 507–517. [[CrossRef](#)]
12. Xiang, H.; Yuan, Y.; Zhao, Y.; Fu, Z. License plate detection based on fully convolutional networks. *J. Electron. Imaging* **2017**, *26*, 145–155. [[CrossRef](#)]
13. Kim, M.; Choi, H.C. Improving Faster R-CNN framework for multi-scale chinese character detection and localization. *IEICE Trans. Inf. Syst.* **2020**, *103*, 1777–1781. [[CrossRef](#)]
14. He, K.; Gkioxari, G.; Dollar, P. Mask R-CNN. In Proceedings of the 2017 International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
17. Gou, C.; Wang, K.Y.; Yao, Y.J.; Li, Z.X. Vehicle license plate recognition based on extremal regions and restricted boltzmann machines. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1096–1107. [[CrossRef](#)]
18. Li, H.; Wang, P.; Shen, C.H. Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1126–1136. [[CrossRef](#)]
19. Xu, Z.B.; Yang, W.; Meng, A.; Lu, N.; Huang, H.; Ying, C.; Huang, L. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In Proceedings of the 2018 European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 255–271.
20. Li, H.; Shen, C.H. Reading car license plates using deep convolutional neural networks and lstms. *arXiv* **2016**, arXiv:1601.05610.
21. Zherzdev, S.; Gruzdev, A. LPRNet: License plate recognition via deep neural networks. *arXiv* **2018**, arXiv:1806.10447.
22. Ghiasi, G.; Fowlkes, C.C. Laplacian pyramid reconstruction and refinement for semantic segmentation. In Proceedings of the 2016 European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 519–534.
23. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the 2018 European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
24. Hsu, G.S.; Chen, J.C.; Chung, Y.Z. Application-oriented license plate recognition. *IEEE Trans. Veh. Technol.* **2013**, *62*, 552–561. [[CrossRef](#)]
25. Selmi, Z.; Alimi, A.M.; Alimi, A.M. Deep learning system for automatic license plate detection and recognition. In Proceedings of the International Conference on Document Analysis and Recognition, Kyoto, Japan, 9–15 November 2017; pp. 1132–1138.
26. Selmi, Z.; Halima, M.B.; Pal, U. DELP-DAR system for license plate detection and recognition. *Pattern Recognit. Lett.* **2020**, *129*, 213–223. [[CrossRef](#)]
27. Molina, M.; Gonzalez-diaz, I.; Diazdemaria, F. Efficient scale-adaptive license plate detection system. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2109–2121. [[CrossRef](#)]
28. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.