Hindawi
Advances in Multimedia
Volume 2022, Article ID 8852142, 10 pages
https://doi.org/10.1155/2022/8852142

Research Article

An Efficient License Plate Detection Approach Using Lightweight Deep Convolutional Neural Networks

Hoanh Nguyen

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

Correspondence should be addressed to Hoanh Nguyen; nguyenhoanh@iuh.edu.vn

Received 12 May 2022; Accepted 1 August 2022; Published 19 August 2022

Academic Editor: Mohammad R. Khosravi

Benefited from deep convolutional neural networks, various license plate detection methods based on deep networks have been proposed and achieved significant improvements compared with traditional methods. However, the high computational cost due to complex structures prevents these methods from being deployed in real-world applications. This paper proposes an efficient license plate detection method based on lightweight deep convolutional neural networks for improving the detection speed. To extract high-level features from input images, this paper designs a lightweight feature pyramid generation module based on a lightweight architecture and depth-wise convolutions. To further enhance feature pyramid, an efficient feature enhancement module is designed to fuse features generated by the region proposal network with backbone features. In the detection network, a light head structure based on fully connected layers is employed to further reduce the computational cost of the model. In experiments, floating point operations and detection ratio are used to evaluate the efficient of the proposed method. Experimental results on public datasets show that the proposed method achieves the best trade-off between speed and accuracy.

1. Introduction

Automatic license plate recognition is a key problem in intelligent transportation systems as it is applied widely in many scenarios, such as highway toll stations or parking lots. An automatic license plate recognition system usually consists of two stages: a license plate detection stage for extracting license plates from input images and a license plate recognition stage for predicting license plate characters. Since license plate detection provides detected license plates for the following license plate recognition, its performance has a huge impact on the performance of the whole system. Most recent state-of-the-art license plate detection approaches are based on deep convolutional neural networks (CNN), where a deep CNN model is first employed to extract features from input images, and a detection head with fully connected layers is then adopted to produce predicted results. Although these approaches achieve great detection performance on public datasets, there are still challenges that prevent them from being deployed in real-world applications. First, these license plate detection approaches are validated on public datasets that contain images capturing under controlled conditions. However, in real-world environments, license plate images might be seriously distorted due to the effect of lightning conditions and occlusions. Second, since these deep CNN license plate detection approaches were conducted in laboratory environments with powerful machines, they are not efficient for real-time applications in real-world scenarios where resources are limited.

In this paper, an efficient license plate detection method based on deep CNN is proposed. For improving the detection speed, ESPNetv2 [1], a lightweight yet high performance network, is used to extract features from input images. Based on the ESPNetv2 structure, this paper designs an efficient feature pyramid generation module which employs depth-wise convolutions and 1 × 1 convolutions to reduce the computational cost. In the detection network, this paper uses a light head structure for producing final detection results. The light head detection network uses proposal boxes generated by a compressed region proposal network (RPN) and feature maps produced by an efficient enhancement module as inputs. Compared with most recent license plate detection methods on public datasets, the proposed method provides significant improvements on the inference speed while achieving comparable detection accuracy. Table 1 provides a summary of features of the proposed method and recent methods for license plate detection to highlight the key contributions of this paper.

The remaining of this paper is organized as follows. Section 2 provides brief reviews of recent license plate detection methods and real-time object detection methods. Section 3 provides theoretical basis of deep CNN. Section 4 introduces the details the proposed framework. Section 5 provides the experimental results and comparisons between the proposed method and other methods on public datasets. Finally, the conclusions are drawn in Section 6.

2. Related Work

2.1. License Plate Detection. With the success of YOLO detection frameworks [2, 10], various approaches have been proposed based on YOLO structures for license plate detection [11, 12]. MD-YOLO [11] proposed a novel model based on YOLO for multidirectional car license plate detection. In this model, a prepositive CNN subnet is first designed to produce attention regions from input images. Then, attention regions are cropped and fed into MD-YOLO branch for determining precise rotational rectangular regions. Laroca et al. [12] introduced a real-time automatic license plate recognition system based on the YOLO detector. The proposed method first detects vehicle regions from input images based on a fast vehicle detection network and then locates license plate positions based on vehicle regions. The authors also designed another two-stage network for character segmentation and recognition. Another approach for license plate detection is to design complex structures for achieving high detection performance. For this purpose, Li et al. [8] proposed a unified deep network for locating license plate positions and recognizing license plate characters. The authors employed VGG-16 structure [13] for extracting features. Proposal regions are produced by a RoI pooling layer based on a region proposal network. For license plate recognition, the authors designed a detection network with fully connected layers to detect license plates and used RNNs to identify plate characters. In [14], a license plate detection method based on two deep networks was designed. A shallow network is first used to remove most of the background regions to reduce the computation cost, and a second deep network is then assigned to detect license plates in the remaining regions. These two networks are trained end-to-end and are complementary to each other to guarantee high detection performance with low computation cost. Recently, Wang et al. [15] proposed Fast-LPRNet based on deep CNN for fast license plate recognition. Fast-LPRNet removes fully connected layers to improve the detection speed. The experiments were implemented on the FPGA hardware. Experimental results show that the Fast-LPRNet achieves high detection accuracy with fast speed. In [9], a license plate detection method based on predicted anchor region proposal network and balanced feature pyramid network was proposed. The predicted anchor region proposal network employs predicted location anchor scheme to generate high-quality sparse proposal boxes. Besides, the balanced feature pyramid network fuses different feature levels to get balanced information from each resolution to improve the detection performance.

2.2. Real-Time Object Detection Methods. Real-time deep CNN object detection is an important issue due to its efficiency when applying in real-world applications. This issue has attracted increasing attention from the research community in recent years. Normally, a real-time object detection framework is based on one-stage object detection structure where predictions are produced by applying a detection head on the backbone feature maps. For example, YOLO [10] and YOLOv2 [2] propose to simplify object detection as a regression problem, which directly predicts object bounding boxes and associated class probabilities without proposal generation stage. Based on this idea, SSD [3] further improves the detection performance by generating predictions of different object scales from different layers of the backbone. DSSD [16] designs deconvolution modules with deconvolution layers to add additional large-scale context to features generated by the backbone network. Different from one-stage structures, two-stage object detection pipelines [17] usually require more computational cost due to their complex structures. To improve the detection speed of two-stage frameworks, recent methods focus on modifying the detection head or adopting a lightweight backbone architecture for extracting features. For this purpose, R-FCN [4] designs fully convolutional architectures to share computations between RoI subnetworks to speed up inference when a large number of proposal boxes are utilized. In [18], a light head detection network was designed to build a fast two-stage object detector. The light head structure includes a single fully connected layer for classification and regression, which greatly reduces the computation cost. In addition, a large-kernel separable convolution is used to produce feature maps with small channels to speed up the detection speed. Recently, ThunderNet [19] proposed a lightweight two-stage generic object detector for real-time object detection. ThunderNet consists of a lightweight backbone for extracting features and a light head detection network for producing final predictions. In addition, a context enhancement module and a spatial attention module are designed to combine feature maps from multiple scales to leverage local and global context information and refine feature distribution.

3. Theoretical Basis of Convolutional Neural Networks

This section provides theoretical basis of CNN, which is mainly used to design the proposed method.

CNN architectures are widely used in the deep learning research community. CNN architectures usually contain three major types of layers: convolution layers, nonlinear

Table 1: Summary of features of the proposed method and recent methods for license plate detection.

Method	Purpose	Based on	Complexity	Accuracy	Advantages	Disadvantages
Tiny-YOLO [2]	Generic object detection	Deep learning	Very low	Low	Fast detection speed; easy to implement	Low accuracy
YOLOv2 [2]	Generic object detection	Deep learning	Medium	High	High accuracy; easy to implement	Struggles to detect small license plates
SSD-300 [3]	Generic object detection	Deep learning	High	High	High accuracy; easy to implement	High computational cost
R-FCN [4]	Generic object detection	Deep learning	High	High	High accuracy; easy to implement	High computational cost
Zhou et al. [5]	License plate detection	Traditional machine learning	High	High	High accuracy; no training process	Low detection speed; cannot detect license plates in difficult conditions
Li et al. [6]	License plate detection	Traditional computer vision	Very high	High	High accuracy; no training process	Very low detection speed; cannot detect license plates in difficult conditions
Yuan et al. [7]	License plate detection	Traditional computer vision	Very low	High	Very high detection speed; no training process	Cannot detect license plates in difficult conditions
Li et al. [8]	License plate detection and recognition	Deep learning	Medium	Very high	Very high accuracy; unified framework for license plate detection and recognition	Low detection speed
Nguyen et al. [9]	License plate detection	Deep learning	High	Very high	Very high accuracy	Low detection speed
Proposed method	License plate detection	Deep learning	Low	Very high	Very high accuracy; fast detection speed	Heavy data augmentation method is needed for training

Features are based on experimental results.

layers, and pooling layers. Convolution layer is the core building block of a CNN architecture as most of the computational budgets happen in convolution layer. Convolution layer employs a kernel (or filter) with learnable weights to convolve with an input feature map to extract its features. There are various types of convolutions, such as standard convolution, depth-wise separable convolution [20], or depth-wise dilated separable convolution [1]. A standard convolution layer convolves a filter $K \in \mathbb{R}^{n \times n \times c \times c'}$ with input features $X \in \mathbb{R}^{W \times H \times c}$ to generate output features $Y \in \mathbb{R}^{W \times H \times c'}$ as follows:

$$Y_{k,l,h} = \sum_{i,j,m} K_{i,j,m,h} . F_{k+i-1,l+j-1,m}, \tag{1}$$

where W and H are the spatial width and height of feature map; c and c' are the number of input and output channels; n is the spatial dimension of the kernel. The number of parameters in a standard convolution layer is $(n.n.c.c')$.

For the purpose of reducing computation cost and model size, depth-wise separable convolution divides a standard convolution into a depth-wise convolution layer and a 1×1 pointwise convolution layer. Depth-wise convolution convolves a filter $K' \in \mathbb{R}^{n \times n \times c}$ with input features $X \in \mathbb{R}^{W \times H \times c}$ to generate output features $Y' \in \mathbb{R}^{W \times H \times c}$ as follows:

$$Y'_{k,h,l} = \sum_{i,j} K'_{i,j,h} . F_{k+i-1,l+j-1,h}. \tag{2}$$

Since depth-wise convolution only filters input channels, pointwise convolution with 1×1 filter is used to compute a linear combination of the output of depth-wise convolution. By combining depth-wise convolution and pointwise convolution, depth-wise separable convolution reduces the computational cost by a factor of α where

$$\alpha = \frac{n.n.c.c'}{n.n.c + c.c'}. \tag{3}$$

Based on depth-wise separable convolution, depth-wise dilated separable convolution replaces depth-wise convolution by depth-wise dilated convolution with a dilation rate of r to filter input channels. By using dilated convolution, depth-wise dilated convolution enables the convolution to learn representations from an effective receptive field of $n_r \times n_r$, where n_r is calculated as follows:

$$n_r = (n-1).r + 1. \tag{4}$$

Depth-wise dilated separable convolution also reduces the computational cost by a factor of α, but it can learn representations from larger receptive fields.

Different from convolution layers, nonlinear layers apply an activation function, such as ReLU or Sigmoid function, on feature maps to enable the modeling of nonlinear functions by the network. On the other hand, pooling layers replace small neighborhood region on a feature map by some statistical information, such as mean or max, about the region to reduce spatial resolution on input features. A deep CNN architecture usually stacks these layers to form a feature pyramid that consists of multiple feature maps with different resolutions. The main computational advantage of a deep CNN architecture is that all the receptive fields in a layer share weights, resulting in a significantly smaller number of parameters than a fully connected architecture.

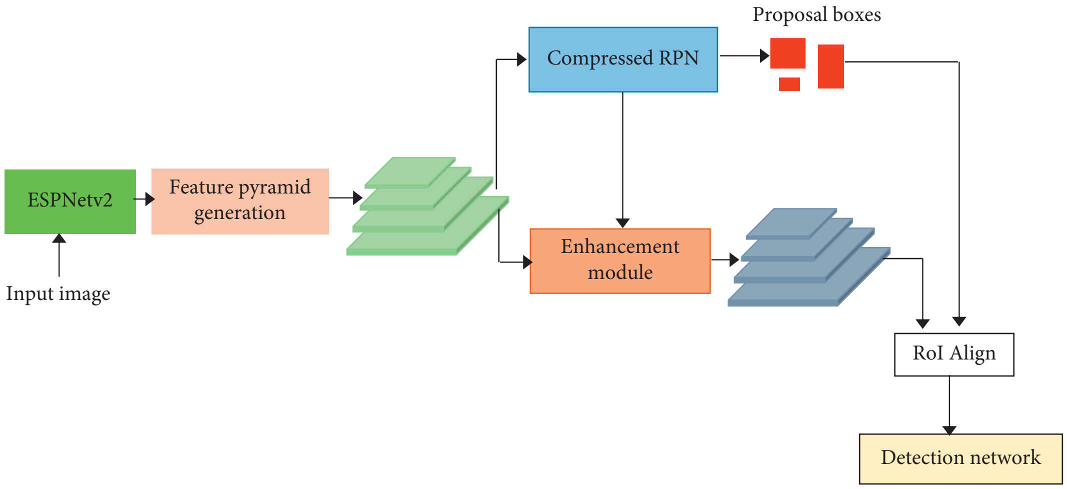

Figure 1: Overall structure of the proposed model.

4. Methodology

The overall structure of the proposed method is shown in Figure 1. Input image is first fed into a lightweight backbone network for extracting high-level semantic feature maps. In order to obtain high detection performance, a feature pyramid generation module is designed based on depth-wise separable convolutions to generate feature pyramid. The compressed RPN then adopts feature pyramid to generate proposal license plates. Meanwhile, intermediate feature layer generated by the compressed RPN is employed to produce enhanced features by a feature enhancement module. Finally, the detection network with a light head structure is designed to generate final predicted results. Moreover, the proposed model utilizes the input resolution at 320 × 320 for both training and testing stages to increasing speed. In the following of this section, each module of the proposed method will be introduced in detail.

4.1. Lightweight Network for Extracting High-Level Semantic Features. Deep object detection models often employ a pretrained deep CNN architecture used for classification task to extract feature maps from input images. For example, VGG [13] was used in faster R-CNN for extracting object features. VGG proposes to design very deep networks using very small convolution filters. Recently, ResNet [21] has been used as feature extraction network in many deep object detectors. ResNet designs residual block which contains a series of layers and a shortcut connection adding the input and output of the block. The residual block is very efficient to build a deeper network. More recently, ResNeXt [22] and ResNest [23] have been designed to improve feature extraction process. Although these above deep CNN architectures extract high-quality object features, they require intensive computations due to large numbers of parameters. To tackle this problem, various methods have been proposed to design a lightweight architecture to reduce the computational cost. For this purpose, MobileNets [20] proposes to use depth-wise separable convolutions to build lightweight deep neural networks. MobileNets significantly reduces the number of parameters when compared to the network using standard convolutions with the same depth in the structure. Different from MobileNets, ShuffleNet [24] introduces pointwise group convolution and channel shuffle to greatly reduce computation cost while maintaining accuracy. Recently, ESPNetv2 [1] proposed to use depth-wise dilated separable convolution to build deep networks. Since depth-wise dilated separable convolution employs large effective receptive fields for extracting features, it achieves better performance compared with depth-wise separable convolution used in MobileNets architecture. For trade-off between the detection performance and computational speed, this paper employs ESPNetv2 [1] as the feature extraction network. Table 2 illustrates the detailed structure of ESPNetv2. As shown in Table 2, ESPNetv2 consists of EESP block and Strided EESP blocks. Based on the ESP block [1] and depth-wise dilated separable convolution which learns object representations from large effective receptive fields while reducing the computational cost, EESP block is designed to reduce complexity of the ESP block. In the EESP block, group point-wise convolutions are first used to project high-dimensional input feature maps into low-dimensional feature maps. Then, 3 × 3 depth-wise dilated separable convolutions are employed in parallel with different dilation rates to learn object representations from a large effective receptive field. Learned feature maps are finally fused in a hierarchical fashion by the computationally efficient hierarchical feature fusion algorithm [25]. In addition to EESP block, Strided EESP block is also designed to add shortcut connection to an input image for downsampling operation. Strided EESP block uses strided depth-wise dilated convolutions to learn object representations efficiently at multiple scales and average pooling to better encode spatial relationships and learn representations efficiently. ESPNetv2 applies EESP and Strided EESP block multiple times at each spatial level to increase the depth of the network except the first spatial level where a standard convolution operation is applied to extract information from input image.

Table 2: The structure of ESPNetv2 used in this paper.

Layer	Type	Output size
Layer 1 (L1)	Standard convolution (3 × 3 × 32, stride 2)	160 × 160
Layer 2 (L2)	1 × strided EESP	80 × 80
Layer 3 (L3)	1 × strided EESP 3 × EESP	40 × 40
Layer 4 (L4)	1 × strided EESP 7 × EESP	20 × 20
Layer 5 (L5)	1 × strided EESP 3 × EESP	10 × 10

In a deep CNN architecture, feature maps at shallow layers contain low-level information, while feature maps at deep layers have high-level information which facilitates the classification process. However, due to low resolution at deep layers, the structure of objects may be destroyed, especially for small objects, thus hindering the detection accuracy of the model. To solve this problem, FPN [26] proposed to combine feature maps at different layers to generate a feature pyramid with high-level information. Benefited from FPN, a great process has been made for deep learning object detection methods, including both one-stage and two-stage object detection methods [27, 28]. Although FPN provides a useful technique for extracting high-quality object representations, its structure involves many extra convolution operations, which increases the number of parameters and computations. For achieving faster speed while maintaining detection accuracy, this paper designs an efficient feature pyramid generation module based on depth-wise convolution [20] which can reduce the computational cost. Figure 2 illustrates the structure of the proposed feature pyramid generation module. The last output layers of ESPNetv2 are denoted as {L2, L3, L4, L5}. Since L2 has high resolution, it is not included in the feature pyramid for computational reason. Instead, this paper applies a global average pooling layer on L5 to generate L6. As a result, {L3, L4, L5, L6} is the final set of feature maps used to produce feature pyramid. For integrating multilevel feature maps, this paper first applies a 1 × 1 convolution on each feature map of the final set to set the number of channels at 256. Then, interpolation and max pooling operations are employed to resize each final feature map to the same size as L4. Once the features are rescaled, the fused feature map is generated by fusing rescaled feature maps through concatenation operations. Based on the fused feature map, output feature maps {O3, O4, O5, O6} are obtained by using upsampling operations and depth-wise convolutions. Specifically, O6 is obtained by applying an upsampling layer and a 1 × 1 convolutional layer, while O3, O4, and O5 are obtained by applying depth-wise convolution layers and 1 × 1 convolutional layers. Since the proposed feature pyramid generation module involves only 1 × 1 convolutions and depth-wise convolutions, it is a computation-friendly pipeline. However, feature maps produced by simple concatenation operations are not optimal, which reduces the performance of the following detection network. To tackle this issue, this paper proposes an enhancement module which will be elaborated in Section 4.2.

4.2. Light Head Detection Network Based on Enhanced Feature Maps. Since feature maps generated by the lightweight feature pyramid generation module are not optimal, it is necessary to enhance these feature maps with more semantic information to improve the performance of the following light head detection network. Based on the observation that the RPN is trained to predict foreground objects, feature maps generated by the first convolution layer in the RPN contain discriminative information between foreground objects and background regions so that they can be used to further enhance feature maps generated by the lightweight feature pyramid generation module. Therefore, this paper proposes a lightweight detection head that produces predictions based on enhanced feature maps generated by an enhancement module and object proposals produced by a compressed RPN. Figure 3 illustrates the structure of the proposed lightweight detection head.

Firstly, this paper modifies the original RPN for improving its processing speed. Specifically, the first 3 × 3 convolution layer in the original RPN is replaced by a 5 × 5 depth-wise convolution layer followed by a 1 × 1 convolution layer with 256 channels (Figure 3). By replacing the traditional convolution layer with a large receptive field depth-wise convolution layer and compressing output channels, the compressed RPN can encode more context information while reducing computational cost.

Secondly, as in FPN [26], this paper also applies the compressed RPN to each level on the output feature pyramid {O3, O4, O5, O6}. At each level, the compressed RPN adopts corresponding input feature map to produce object proposals. For predefined anchor boxes, due to small size of license plates in natural scene images, this paper defines anchor box with areas of {5, 15, 30, 60} on {O3, O4, O5, O6}, respectively. Each anchor box is associated with one aspect ratio (width/height = 5) due to rectangular shape of license plates. Since proposals are usually overlap each other, Soft-NMS [29] is used to suppress overlapped proposals. At the same time, a feature enhancement module is design to enhance each feature level on the output feature pyramid (Figure 3). In the feature enhancement module, feature maps in the compressed RPN which contain discriminative information between foreground objects and background regions are employed to generate enhanced feature maps. To fuse feature maps generated by the compressed RPN and input feature maps, a 1 × 1 convolution layer followed by a sigmoid layer is used. Here, 1 × 1 convolution layer is used to compares output channels while sigmoid layer is adopted to constrain output values within [0, 1]. Since the feature enhancement module involves only 1 × 1 convolutions, its computational cost is negligible.

Finally, benefited from the powerful feature maps generated by the feature enhancement module, this paper

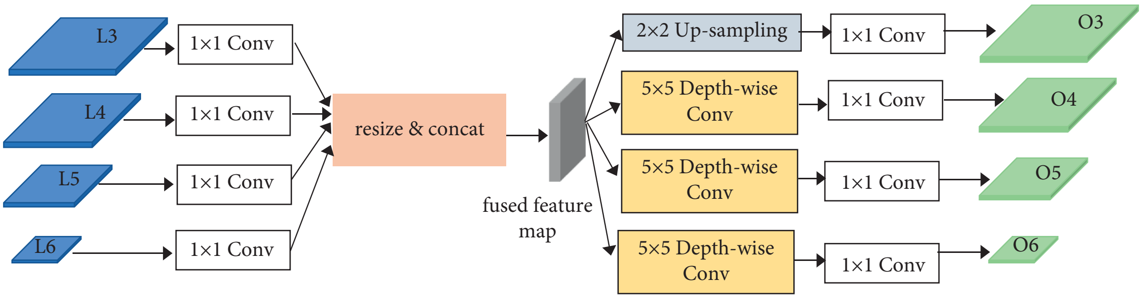

Figure 2: The structure of the proposed feature pyramid generation module.

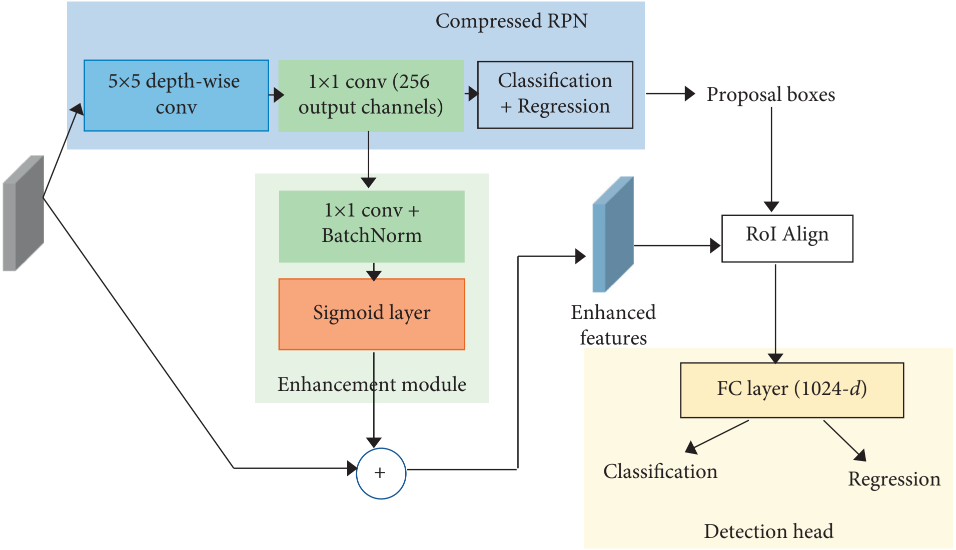

Figure 3: The lightweight detection head with the compressed RPN and the feature enhancement module.

follows light-head R-CNN structure [18] to design a lightweight detection head (Figure 3). Specifically, instead of using two fully connected layers followed by two parallel fully connected layers for classification and regression as in R-CNN [30], this paper applies a single 1024-d fully connected layer to reduces the computational cost of the detection head without sacrificing accuracy. In addition, the RoI align scheme [31] is employed to extract high-quality object features for classification and regression based on object proposals and input feature maps. For assigning a proposal to appropriate feature level, this paper uses its size to choose the best feature level as in [26]. With the lightweight structure in the detection head, the proposed model can strike the best trade-off between speed and accuracy.

5. Results and Discussion

5.1. Datasets. To evaluate the detection performance and prove the robustness of the proposed method, this paper conducts experiments based on two widely used public license plate datasets: AOLP [32] and PKU vehicle dataset [7]. Table 3 provides a summary of the two datasets.

AOLP dataset [32]: the dataset consists of 2049 images containing Taiwan license plates. Images in the dataset were captured in different weather conditions, locations, and time. For evaluating the detection performance of license plate methods in different scenarios, images in this dataset are grouped into three subsets: access control (AC), traffic law enforcement (LE), and road patrol (RP). AC subset contains 681 images captured when a vehicle traverses a fixed passage with a significantly lower speed than normal or comes to a full stop. On the other hand, LE subset contains 757 images captured by a roadside camera when a vehicle violates traffic laws. Therefore, images in the LE subset may have heavily cluttered backgrounds with multiple road signs or pedestrians. Finally, RP subset contains 611 images captured by a camera mounted on a patrol car. As a result, images in the RP subset may have arbitrary viewpoints and distances. Since there is no standard split for AOLP dataset, this paper employs AC and LE subsets as the training sets and uses RP as the testing set.

PKU vehicle dataset [7]: the dataset contains 3828 images with Chinese car license plates captured in different scenarios. Based on capturing environment, images in the dataset are divided into five groups (G1–G5). Specifically, while images in G4 were captured during nighttime, images in other groups were captured during daytime. In addition, images in G1, G2, and G3 were taken on highways, while images in G4 and G5 were taken on city roads and at intersections with crosswalks, respectively. Moreover, there is

Table 3: The summary of license plate datasets used in this paper.

Datasets	Subsets	Number of images	Number of license plates	Year	License plate distance
AOLP dataset	AC	681	681	2012	Near
	LE	757	757		
	RP	611	611		
PKU vehicle dataset	G1	810	810	2016	Far
	G2	700	700		
	G3	743	743		
	G4	572	572		
	G5	1152	1438		

Table 4: Detection results on the RP subset of AOLP dataset.

Model	Feature extraction network	Input resolution	Detection ratio (%)	MFLOPs
Tiny-YOLO [2]	Tiny darknet	416 × 416	65.5	3490
YOLOv2 [2]	Darknet-19	416 × 416	81.8	17400
SSD-300 [3]	VGG-16	300 × 300	83.5	31750
R-FCN [4]	ResNet-50	600 × 1000	84.3	58900
Proposed method	ESPNetv2	320 × 320	99.35	5210

one license plate for each image in G1, G2, G3, and G4, while images in G5 contain multiple license plates. As there is no standard split for this dataset, this paper employs CarFlag-Large dataset [8], which contains 460000 images containing Chinese license plates, to train the proposed network.

5.2. Implementation Details. The proposed method is implemented based on Pytorch framework with NVIDIA Titan X GPU. All input images for both training and testing stages are resized to 320 × 320. This paper employs pretrained ESPNetv2 model trained on the ImageNet dataset. The proposed model is trained end-to-end using synchronized SGD with a weight decay of 0.0001 and a momentum of 0.9. The batch size is set to 8 for single GPU. Since the proposed model uses images with low resolution for training, heavy data augmentation method [3] is employed to improve training results. The proposed model is trained for 50 K iterations on AOLP dataset and 100 K iterations on CarFlag-Large dataset. The learning rate is set at 0.001 and decays by a factor of 0.1 at 50% and 75% of the total iterations.

For evaluation metrics, as there is no uniform criterion for evaluating the detection performance of different license plate detection methods, this paper employs the detection ratio [7] as evaluation metric. Specifically, for each detected result, only when the IoU value between the detected license plate and the ground truth is greater than 50%, the detected license plate is considered to be correct. Here, the IoU value is defined in (5), where DL denotes the area of the detected license plate and GL represents the area of ground truth license plate. Furthermore, this paper compares the inference speed of different license plate detection methods to validate the low computational complexity of the proposed method.

$$IoU = \frac{DL \cap GL}{DL \cup GL}. \tag{5}$$

5.3. Detection Results on AOLP Dataset. To prove the efficient of the proposed method, this paper compares the detection performance of the proposed model with that of recent fast object detectors, including Tiny-YOLO [2], YOLOv2 [2], SSD-300 [3], and R-FCN [4]. These generic object detectors were designed for fast detection speed based on lightweight structures. The results are shown in Table 4. The proposed model obtains the best detection ratio on the RP subset of AOLP dataset. Specifically, the proposed model improves detection ratio by 33.85 points, 17.55 points, 15.85 points, and 15.05 points compared with Tiny-YOLO, YOLOv2, SSD-300, and R-FCN, respectively. For evaluating the complexity of the proposed model, this paper uses floating point operations (FLOPs) to calculate the computational cost of different models on AOLP dataset. Results are shown in Table 4. All results are reported based on mmdetection codebase [33]. By using a lightweight feature extraction network and a light head detection network, the proposed model uses fewer FLOPs compared to YOLOv2, SSD-300, and R-FCN. Compared with tiny-YOLO, the proposed model acquires higher FLOPs. However, the proposed model achieves significant higher detection ratio than tiny-YOLO. It should be noted that although the proposed model employs low resolution input images for both training and testing, it achieves better detection performance than methods used high resolution input images. This result shows that the proposed lightweight feature extraction network and feature enhancement module are very efficient for producing high-level semantic information feature maps which improve the performance of the subsequent detection subnet.

5.4. Detection Results on PKU Vehicle Dataset. For PKU vehicle dataset, this paper compares the detection results of different license plate methods as shown in Table 5. In Table 5, methods proposed by Li et al. [8] and Nguyen [9] are based on deep networks, while methods designed by Zhou et al. [5], Li et al. [6], and Yuan et al. [7] employ handcrafted features and a traditional classifier to locate license plates. It is obvious from Table 5 that these recent methods using deep

Table 5: Detection results on the PKU vehicle dataset.

Method	Detection ratio (%)					Inference time (ms)
	G1	G2	G3	G4	G5	
Zhou et al. [5]	95.43	97.85	94.21	81.23	82.37	475
Li et al. [6]	98.89	98.42	95.83	81.17	83.31	672
Yuan et al. [7]	98.76	98.42	97.72	96.23	97.32	42
Li et al. [8]	99.88	99.71	99.46	99.83	98.68	283
Nguyen [9]	99.88	99.86	99.73	99.83	99.41	420
Proposed method	99.38	99.71	99.73	99.65	98.68	72

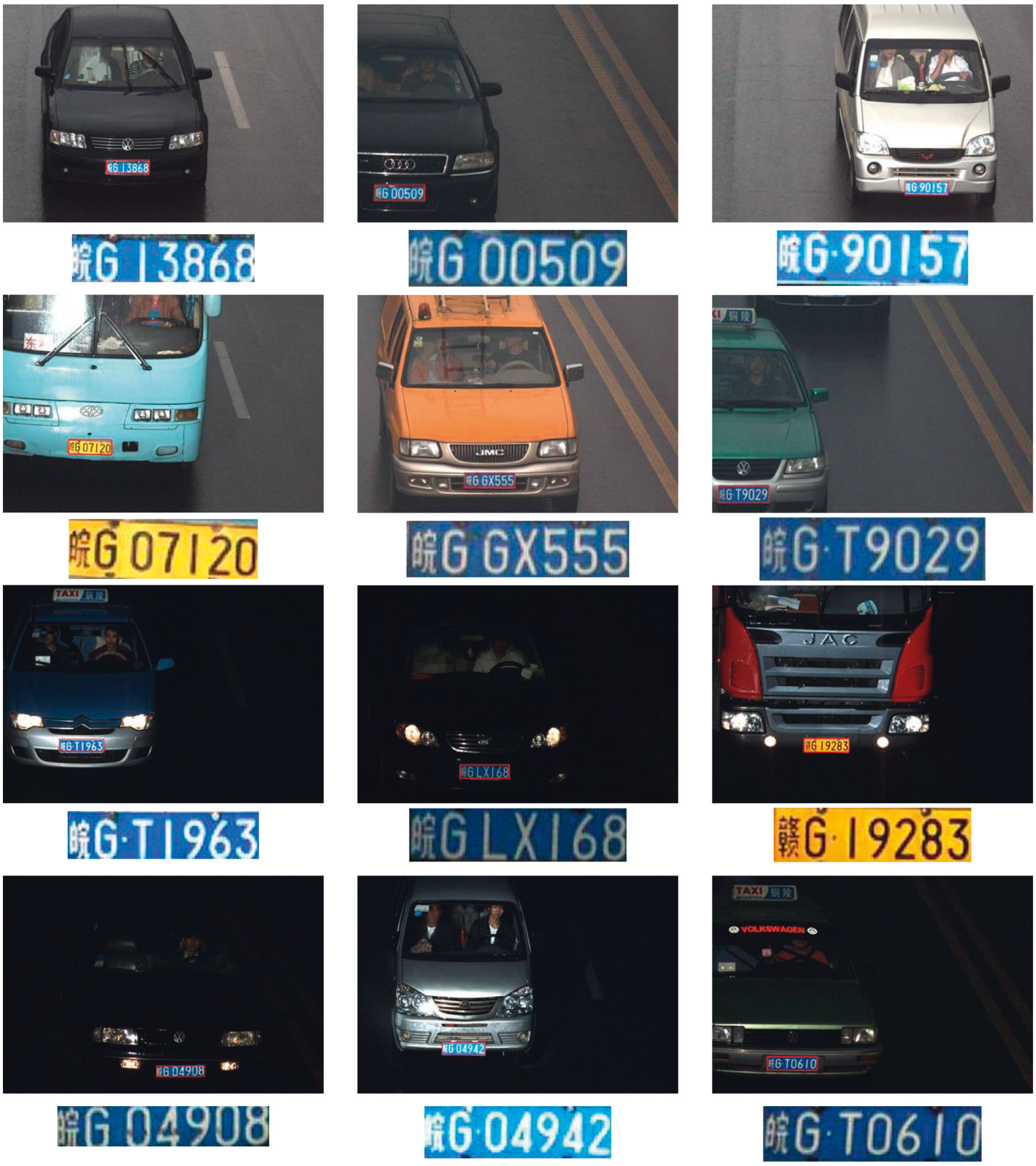

Figure 4: Examples of detection results of the proposed method on PKU vehicle dataset.

networks achieve better detection accuracy than methods using hand-crafted features. Comparison results in Table 5 show that the proposed method obtains comparable detection accuracy on all subsets compared to other deep CNN-based methods (i.e., methods proposed by Li et al. [8] and Nguyen [9]). However, the proposed model significantly improves the detection speed. Specifically, the proposed model takes 72 ms for processing an image, while methods proposed by Nguyen [9] requires 420 ms. Method proposed by Yuan et al. [7] is faster than the proposed method since it uses simple linear SVMs. However, the proposed method obtains better detection accuracy on all five subsets. The results show the efficiency of the proposed method in terms of detection accuracy and inference speed. Figure 4 shows some examples of detection results of the proposed method on PKU vehicle dataset.

6. Conclusions

This paper proposes an efficient license plate detection method based on lightweight structures. For feature extraction, a lightweight feature extraction network is designed to produce high-level semantic information feature maps from input images. The feature extraction network includes a deep backbone network to extract base feature maps and an efficient feature pyramid generation module to strengthen the base features. In the detection stage, a light head detection network is employed to produce final predictions. The light head detection network adopts proposal boxes produced by a compressed RPN and enhanced feature maps generated by an enhancement module as inputs. Experimental results on two public datasets show that the proposed method achieve the best trade-off between speed and accuracy.

Data Availability

The codes used in this paper are available from the author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "Espnetv2: a light-weight, power efficient, and general purpose convolutional neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9190–9200, Long Beach, CA, USA, June 2019.

[2] J. Redmon and F. Ali, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[3] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Cham, September 2016.

[4] J. Dai, Yi Li, K. He, and J. Sun, "R-fcn: object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 379–387, Barcelona, Spain, December 2016.

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

[6] Bo Li, B. Tian, Y. Li, and D. Wen, "Component-based license plate detection using conditional random field model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1690–1699, 2013.

[7] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A robust and efficient approach to license plate detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1102–1114, 2017.

[8] H. Li, P. Wang, and C. Shen, "Toward end-to-end car license plate detection and recognition with deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, 2019.

[9] H.. Nguyen, "Predicted anchor region proposal with balanced feature pyramid for license plate detection in traffic scene images," *Complexity*, vol. 2020, pp. 1–11, Article ID 5137056, 2020.

[10] J. Redmon, S. Divvala, R. Girshick, and F. Ali, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[11] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 507–517, 2018.

[12] R. Laroca, E. Severo, L. A. Zanlorensi et al., "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proceedings of the 2018 International Joint Conference on Neural Networks (Ijcnn)*, pp. 1–10, IEEE, Rio de Janeiro, Brazil, July 2018.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," p. 1556, 2014, https://arxiv.org/abs/1409.1556.

[14] Li Zou, M. Zhao, Z. Gao, M. Cao, H. Jia, and M. Pei, "License plate detection with shallow and deep CNNs in complex environments," *Complexity*, vol. 2018, pp. 1–6, Article ID 7984653, 2018.

[15] Z. Wang, Yu Jiang, J. Liu, S. Gong, J. Yao, and F. Jiang, "Research and implementation of fast-LPRNet algorithm for license plate recognition," *Journal of Electrical and Computer Engineering*, vol. 2021, pp. 1–11, Article ID 8592216, 2021.

[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, C. Alexander, and Berg, "Dssd: deconvolutional single shot detector," 2017, https://arxiv.org/abs/1701.06659.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[18] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: in defense of two-stage object detector," 2017, https://arxiv.org/abs/1711.07264.

[19] Z. Qin, Z. Li, Z. Zhang et al., "ThunderNet: towards real-time generic object detection on mobile devices," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6718–6727, Seoul, Korea, October 2019.

[20] A. G. Howard, M. Zhu, Bo Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, Honolulu, HI, USA, July 2017.

[23] H. Zhang, C. Wu, Z. Zhang et al., "Resnest: split-attention networks," 2020, https://arxiv.org/abs/2004.08955.

[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: an extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Salt Lake City, UT, USA, June 2018.

[25] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, “Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 552–568, Cham, October 2018.

[26] T.-Yi Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, December 2017.

[27] Z. Tian, C. Shen, H. Chen, and He Tong, “Fcos: fully convolutional one-stage object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9627–9636, April 2019.

[28] S. Liu, Qi Lu, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, March 2018.

[29] N. Bodla, B. Singh, R. Chellappa, S. Larry, and Davis, “Soft-NMS--improving object detection with one line of code,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5561–5569, Venice, Italy, October 2017.

[30] R.. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[31] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.

[32] G.-S. Hsu, J.-C. Chen, and Yu-Zu Chung, “Application-oriented license plate recognition,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, 2013.

[33] K. Chen, J. Wang, J. Pang et al., “MMDetection: open mmlab detection toolbox and benchmark,” 2019, https://arxiv.org/abs/1906.07155.