electronics

MDPI

Article

A Lightweight License Plate Recognition Method Based on YOLOv8

Xingwei Zhang [1] and Shenglin Yu [2,*]

[1] School of Information Science and Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202312490481@nuist.edu.cn
[2] School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
* Correspondence: 002201@nuist.edu.cn; Tel.: +86-18651851092

Abstract

To address the challenges faced by license plate recognition systems in certain scenarios—such as complex plate backgrounds, plate tilt, and the large model size that hinders deployment on resource-constrained devices—this paper proposes a compact and computation-efficient license plate detection algorithm that maintains the required recognition accuracy while being easy to deploy on edge computing devices. Experimental results show that the optimized detection model reduces network parameters by approximately 32% without compromising accuracy, while the model file size is similarly reduced by about 32%, significantly conserving device resources. For the recognition stage, LPRNet is further optimized. Experiments demonstrate that the improved recognition network achieves a 1.2% higher accuracy compared to the baseline, with almost no increase in model size, thereby delivering better license plate recognition performance. The combined detection and recognition models occupy less than 6 MB of storage, offering clear advantages in recognition rate, robustness, and resource-efficient design, making them well-suited for deployment on edge devices.

Keywords: object detection; license plate recognition; YOLOv8; LPRNet

check for updates

Received: 24 July 2025
Revised: 17 August 2025
Accepted: 28 August 2025
Published: 31 August 2025

Citation: Zhang, X.; Yu, S. A Lightweight License Plate Recognition Method Based on YOLOv8. *Electronics* **2025**, *14*, 3482. https://doi.org/10.3390/electronics14173482

Copyright: © 2025 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

Intelligent parking has become an important part of China's smart city construction. License plate recognition is the main means for intelligent transportation systems to conduct vehicle statistics and control. Currently, the process of license plate recognition tasks usually consists of two parts: license plate location detection and license plate recognition. With the development of technology, license plate location methods can be roughly divided into two categories: traditional and modern. Traditional license plate location methods have some disadvantages, such as poor adaptability to environmental changes, poor robustness to license plate deformation and angle changes, and high false detection rate in some scenarios where the background and license plate color are similar. Modern license plate location methods are achieved by learning about the target through deep learning networks, which solves some disadvantages of traditional license plate location methods and significantly improves the accuracy and robustness of license plate detection in various complex scenarios. At the same time, license plate recognition can be divided into two types: traditional recognition methods are based on image processing methods and optical character recognition (OCR) to jointly identify, with low robustness and low

recognition rate for blurry and tilted license plates. Meanwhile, the recognition accuracy of these methods comes from the quality of the image and the segmentation of characters, and the quality of segmentation also affects the recognition effect. Modern recognition methods are based on deep learning methods, with strong robustness, the ability to adapt to various complex scenarios, and high recognition accuracy. Currently, most of the recognition models with high accuracy and robustness are based on deep neural networks. In the implementation of license plate detection and recognition. Xu, G.Z [1] proposed a precise license plate location algorithm that can simultaneously output rectangular target detection boxes and the exact four points of a real license plate, aiming to solve the problem that the rectangular detection boxes output by the target detection algorithm of the deep learning technology do not fit well with the actual license plate area, affecting the accurate positioning of the license plate. He, Z.L [2] proposed an improved license plate location algorithm for YOLOv3, which reduces the model size while improving the recognition accuracy. R, W.J [3] proposed an end-to-end intelligent license plate recognition method based on a deep neural network model, using the YOLOv3 network to locate the license plate, then correcting the tilted license plate through perspective changes, and sending the corrected license plate to the license plate recognition network to improve the accuracy of character recognition. However, this method still has room for improvement in terms of detection accuracy and speed. G, S.X [4] used UNet as an image enhancement network and LPRNet as a license plate recognition network, and deployed the trained model on a Raspberry Pi, achieving a high license plate recognition accuracy, but the speed was relatively slow. Wu, H.W [5] used deep learning models such as YOLOv5, U2-net, and LPRNet to implement license plate detection and recognition in some complex scenarios and achieved good recognition results, but the LPRNet model in this method had not achieved satisfactory recognition results for some length-uncertain license plates and partially blurry license plates, and the YOLOv5 model contained a large number of convolution operations, resulting in a relatively large generated model, which faced difficulty in meeting the application requirements of embedded systems.

In conclusion, for current license plate recognition tasks, this study addresses several key issues with existing models, including their large size and difficulty in deployment on edge devices, limited robustness in recognizing license plates, and low recognition accuracy. The proposed improvements focus on two main aspects: the compact and computation-efficient design of license plate detection and recognition algorithms, and the refinement of character recognition methods. Specifically, an improved YOLOv8n is adopted for license plate localization, while an enhanced mainstream convolutional neural network (LPRNet) is employed for character recognition, with targeted optimizations applied to LPRNet. With this resource-efficient design, the improved YOLOv8n–LPRNet cascaded network effectively captures sequential information from images, accurately recognizes continuous text, and delivers strong license plate recognition performance.

The principal contributions of this research are as follows:

(1) We propose a novel network structure termed GCE. In the GCE module, the input feature map is divided channel-wise into two sub-feature maps, denoted as A and B. The A sub-feature map is processed through multiple convolutional layers, and the resulting features are concatenated with the B sub-feature map. The merged feature map is then refined using a squeeze-and-excitation (SE) attention mechanism, enhancing the extraction of key target features while suppressing irrelevant noise. This design enables the model to achieve more accurate target recognition, while maintaining accuracy through a compact and computationally efficient architecture.

(2) In YOLOv8, we incorporated the BIFPN structure for multi-scale feature fusion. Its core innovation lies in bidirectional cross-scale connections and weighted feature fusion, which efficiently integrate feature information across different levels, yielding improved performance in small target detection. Its unique feature fusion mechanism endows it with strong adaptability for license plate localization in complex environments. Specifically, its learnable weight mechanism enables the network to autonomously select features critical for license plate localization—such as license plate border textures and character contours—thereby suppressing background noise and enhancing the model's license plate detection capability.

(3) Building upon the traditional LPRNet architecture, we propose an improved model, N_E_LPRNet, which features a reduced computational load and can be deployed on mobile devices with limited resources. In neural networks, the dropout mechanism is commonly employed to prevent overfitting by randomly deactivating neurons. However, given that the original LPRNet model has relatively low complexity and the dataset used contains a large number of samples, overfitting is rarely observed. Moreover, the inherent randomness of dropout may lead to unstable recognition results due to neuron pruning. To address this, the improved model replaces dropout layers with batch normalization (BN) layers, which not only accelerates training but also enhances the model's generalization ability and robustness. In addition, an Efficient Multi-scale Attention (EMA) module is incorporated into the network to strengthen its capacity for extracting and learning discriminative features from license plate images, thereby improving recognition accuracy.

2. Proposed Methods

2.1. YOLOv8n Algorithm Architecture

YOLOv8 is a target detection algorithm developed by Ultralytics. It has made various improvements and optimizations based on previous versions, significantly enhancing the performance of target detection. It has achieved remarkable improvements in speed and accuracy, making it suitable for a wider range of application scenarios. This algorithm has advantages such as high accuracy, fast speed, and strong generalization ability. The YOLOv8 network consists of three core components: the backbone network (Backbone), the neck (Neck), and the head (Head). The backbone network is responsible for extracting feature information from the image, while the neck compresses and fuses the features extracted by the backbone network. Finally, the processed feature maps are sent to the head network to perform target localization and classification tasks [6].

Due to the large number of convolution operations in the YOLOv8 model, the resulting target detection network is not well-suited for deployment on certain resource-constrained mobile devices. To address this issue, we design a mobile-friendly target detection model based on YOLOv8. Specifically, we optimize YOLOv8 through model compactness and parameter reduction, and integrate a compact and computation-efficient GCE module, which substantially decreases the number of network parameters and improves computational efficiency, making the network well-adapted for deployment on mobile platforms.

Given the diversity of license plate detection datasets and the complexity of license plate backgrounds, the original network may suffer reduced positioning accuracy, posing challenges for license plate detection. To address the accuracy degradation caused by complex backgrounds, this study optimizes the multi-scale feature fusion component in the neck of YOLOv8. Specifically, we introduce an efficient bidirectional cross-scale connection and a weighted feature fusion strategy, replacing the original PANet architecture in YOLOv8 with the Bidirectional Feature Pyramid Network (BiFPN), which offers superior performance in detecting small targets such as license plates. This modification enhances

the model's ability to accurately localize license plates of varying sizes and orientations, thereby improving overall detection accuracy.

Traditional YOLO-based license plate detection methods localize targets using bounding boxes aligned parallel to the image borders. For targets that are not parallel, this approach introduces additional background noise, which can negatively impact the performance of subsequent license plate recognition networks. To address this, the license plate localization algorithm proposed in this study not only predicts the bounding rectangle containing the license plate but also estimates the coordinates of its four vertices, enabling precise cropping based on these points. In practical scenarios, some license plates may appear tilted; therefore, a perspective transformation is applied to correct the tilt before passing the result to the LPRNet network, thereby improving the accuracy of the character recognition stage. Consequently, the detection head of the license plate localization network adopts a pose detection head rather than the traditional detection head. The overall architecture of the improved GCE_B_YOLO network is illustrated in Figure 1.

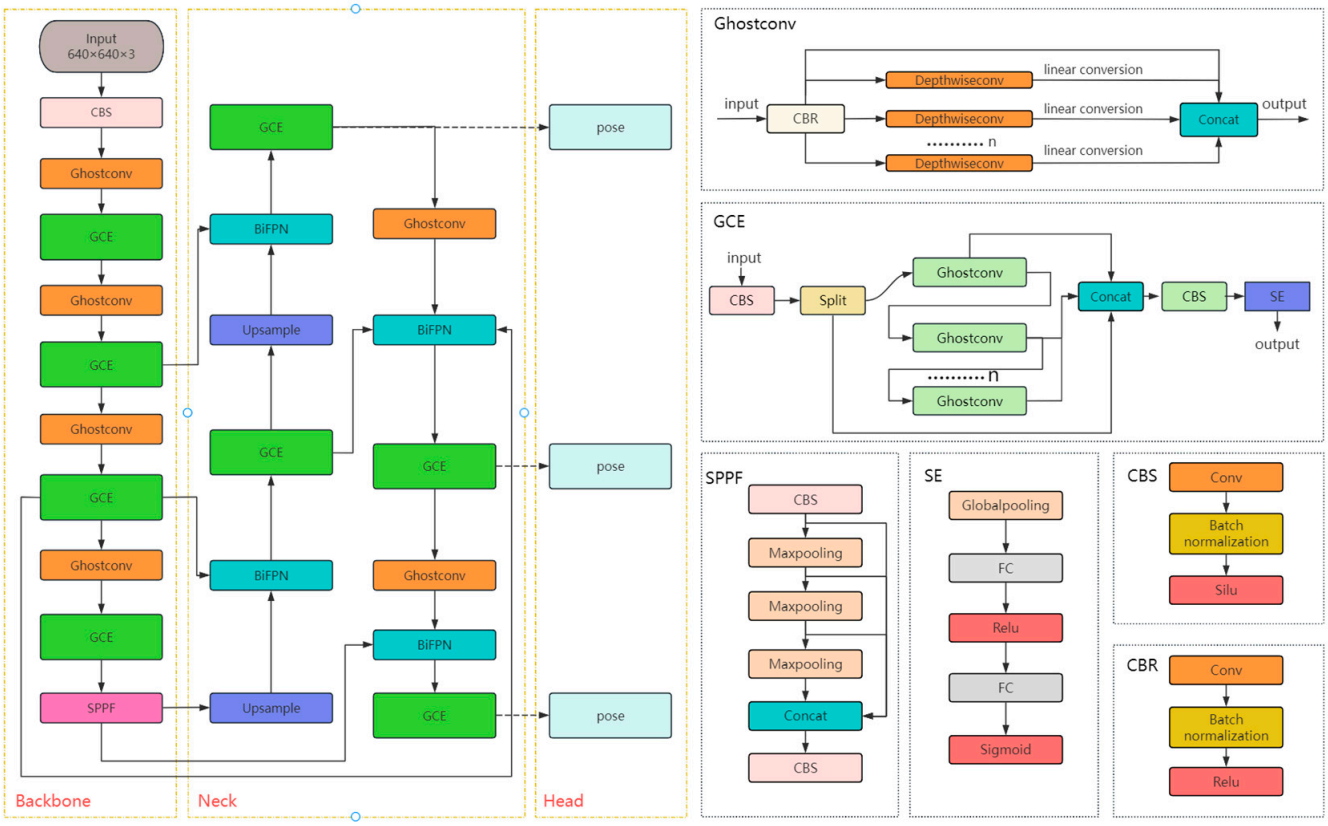

Figure 1. GCE_B_YOLO network structure.

2.2. GCE Module

The GCE module is designed based on the Cross Stage Partial (CSP) structure, an architectural approach in deep learning that optimizes feature extraction and computational efficiency. The core principle of CSP is to split and recombine feature maps across network stages, thereby reducing computational cost while maintaining or even enhancing model performance. Specifically, the CSP structure divides the feature map into two parts to improve the network's nonlinear representation capability, enabling it to better handle complex image features. In the GCE module, after the input feature map passes through the first convolutional layer, it is split into two branches, each processed by different convolutional layers before being merged. This design allows the model to capture richer contextual information, improving target localization accuracy. The merged feature map is

subsequently processed by a second convolutional layer and then output. Through this mechanism, the network effectively captures complex image features, yielding improved performance in object detection tasks. Moreover, this structure not only reduces inference time and simplifies the network design but also significantly decreases the number of parameters while maintaining competitive accuracy, resulting in a substantial reduction in computational load (FLOPs).

There are various approaches to achieving model compactness and computational efficiency. For instance, MobileNet [7] introduced depthwise separable convolution to create a more compact and computation-efficient backbone network. In Zhang et al. [8], incorporating depthwise convolution (DWConv) into the residual modules of the backbone substantially reduced the number of model parameters. However, because DWConv processes information independently across channels, it may result in the loss of key semantic features, thereby weakening the model's feature extraction capability and reducing accuracy [9]. In this paper, we adopt another parameter-efficient convolutional operation, namely the Ghost convolution module. Unlike traditional convolution, which applies convolutional operations to all channels of the input feature map—requiring a large number of parameters—Ghost convolution combines standard convolution with inexpensive linear transformation operations. Specifically, conventional convolution is used to generate part of the intermediate feature maps, while the remaining feature maps are generated through linear transformations and then merged with those from conventional convolution to approximate the complete feature representation. This design substantially reduces computational cost and the number of parameters compared to standard convolution. The main structure of Ghost convolution is illustrated in Figure 2.

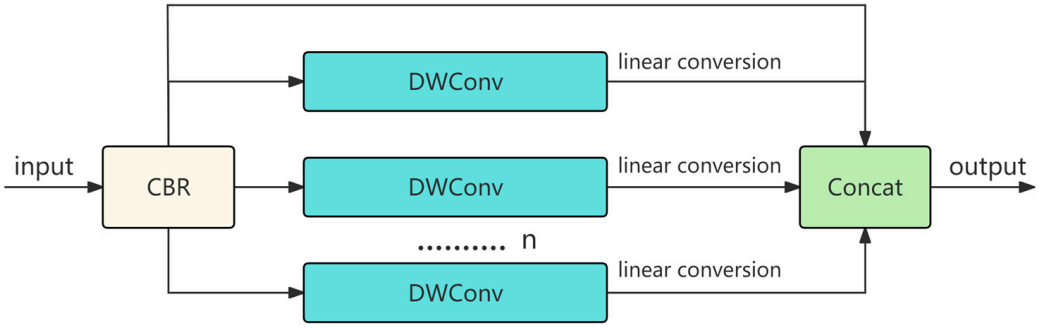

Figure 2. Ghost convolutional structure.

From the structure diagram, it can be seen that the computational complexity (*FLOPs*1) of GhostConv is as shown in formula (1), and the parameter quantity (*Prams*1) is as shown in formula (2):

$$FLOPs1 = c \times k^2 \times \left(\frac{n_0}{s}\right) \times h_0 \times w_0 + \left(\frac{n_0}{s}\right) \times d^2 \times (s-1) \times h_0 \times w_0 \quad (1)$$

$$Prams1 = c \times k^2 \times \left(\frac{n_0}{s}\right) + \left(\frac{n_0}{s}\right) \times d^2 \times (s-1) \quad (2)$$

The computational complexity (*FLOPs*2) of the ordinary convolution is shown in Equation (3), and the number of parameters (*Prams*2) is shown in Equation (4):

$$FLOPs2 = c \times k^2 \times n_0 \times h_0 \times w_0 \quad (3)$$

$$Prams2 = c \times k^2 \times n_0 \quad (4)$$

The computational cost of GhostConv is compared with that of ordinary convolution as shown in formula (5):

$$Computational\ load\ ratio = \frac{FLOPs1}{FLOPs2} = \frac{c \times k^2 \times \frac{1}{s} + \frac{1}{s} \times d^2 \times (s-1)}{c \times k^2} \approx \frac{1}{s} \quad (5)$$

The parameter compression ratio calculation of GhostConv compared to ordinary convolution is shown in formula (6):

$$Parameter\ ratio = \frac{FLOPs1}{FLOPs2} = \frac{c \times k^2 \times \frac{1}{s} + \frac{1}{s} \times d^2 \times (s-1)}{c \times k^2} \approx \frac{1}{s} \quad (6)$$

In the above formula, "c" represents the number of channels of the input feature map, "h_0" represents the output height of the feature map, "w_0" represents the output width of the feature map, "n_0" represents the output channel number of the feature map, "k_0" represents the size of the ordinary convolution kernel, "d" represents the convolution kernel size for extracting linear operations of the image, "$(s)(s << n)$" represents the number of times of linear transformation of the linear feature map, and through the calculation of formulas (5) and (6), it can be obtained that the compression ratio of the computational and parameter quantities of the ordinary convolution and Ghostconv approaches infinity as "(s)", indicating that using Ghostconv convolution to replace the ordinary convolution can indeed reduce the parameters and computational quantities of the model.

The SE attention module can enhance the channel features of the input feature map without changing the size of the input feature map. It is often applied in visual models and can be used directly. The SE module explicitly models the dependencies between channels, enabling the network to adaptively enhance important features and suppress redundant information. The main purpose of the SE module is to improve the model's sensitivity to channel features. This module is lightweight and can be applied in existing network structures, requiring only a small increase in computational cost to achieve performance improvement. The SE module consists of three core operations: Squeeze (compression), Excitation (stimulation), and Scale (re-scaling). Its structure is shown in Figure 3:

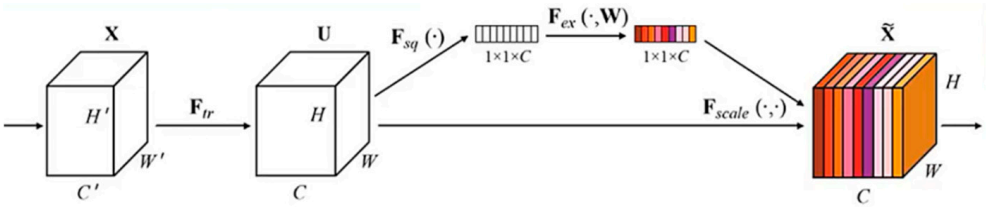

Figure 3. SE attention module structure.

The CSP structure enables spatial feature fusion across network stages. In the context of license plate localization, this structure effectively integrates edge information from license plates of different scales and the texture and color information of text regions, facilitating accurate boundary localization for both license plates and characters. The Ghost convolution module substantially reduces the model's computational complexity, while the SE attention mechanism enhances feature representation along the channel dimension. For example, it can strengthen the blue channel in blue license plates and the gradient channel along text edges, while suppressing background interference, thereby highlighting key regions and improving the model's robustness in complex background scenarios. Based on the functionalities of these components, we designed a novel network structure, illustrated in Figure 4. This design maintains a high level of accuracy while significantly reducing the number of model parameters.

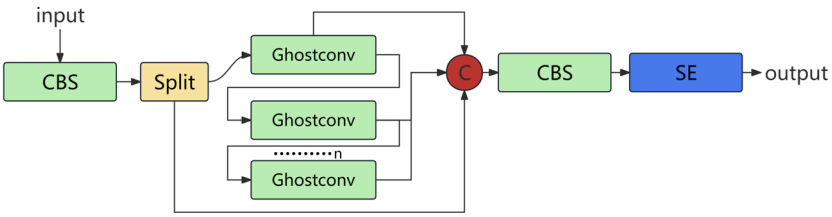

Figure 4. GCE network structure.

2.3. BIFPN Module

In YOLOv8, the neck network employs PAN (Path Aggregation Network) to perform bidirectional fusion of multi-scale features, effectively enhancing the model's capability to detect targets of varying scales. However, this conventional direct concatenation fusion method exhibits notable limitations when handling small targets, such as license plates. Specifically, PAN constructs a bidirectional feature transfer mechanism: the top-down path conveys high-level semantic information (e.g., the overall representation of the vehicle), while the bottom-up path leverages cross-layer connections to transmit low-level detailed features (e.g., the edge texture of the license plate). In the absence of an adaptive weighting mechanism during feature fusion, high-level semantic features tend to dominate, leading to suppression of the detailed features critical for small target detection. This, in turn, reduces the detection accuracy for small objects such as license plates. To address this limitation, this study integrates a Bidirectional Feature Pyramid Network (BiFPN) into the neck, enabling more effective multi-scale feature fusion and improving the model's detection performance for small targets.

BIFPN (Bidirectional Feature Pyramid Network) is a feature fusion architecture optimized for object detection. Its core innovation lies in bidirectional cross-scale connections and weighted feature fusion, enabling efficient integration of feature information across multiple levels. BIFPN has demonstrated strong performance in small target detection. Its feature fusion mechanism provides robust adaptability for license plate localization in complex environments. Through a learnable weighting mechanism, the network can automatically prioritize critical features for license plate positioning, such as border textures and character contours, while suppressing background interference. In challenging scenarios, such as tilted license plates captured by cameras, the multi-scale feature interactions in BIFPN effectively capture invariant characteristics—such as character proportions and spatial arrangements—thereby enhancing the stability and accuracy of license plate localization. The network structure of BIFPN is illustrated in Figure 5.

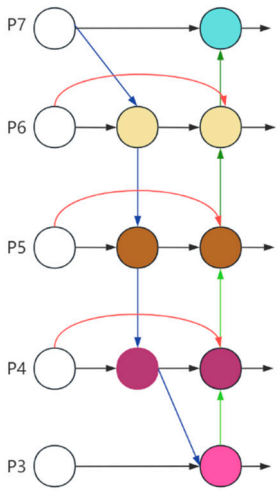

Figure 5. BIFPN network structure.

2.4. Improvement of the LPRNET Network Model

Currently, there are various algorithms for license plate recognition. Among them, ResNet101 [10] achieves relatively high recognition accuracy but entails a large computational load during inference. In contrast, SqueezeNet [11] has a lower computational cost but suffers from limited recognition accuracy. The LPRNet model achieves recognition accuracy second only to ResNet101 while maintaining a model size of only 1.7 MB and low computational complexity. Therefore, this paper adopts LPRNet with a compact and computation-efficient network architecture as the primary algorithm for license plate recognition. Unlike traditional methods that rely on manual feature extraction and complex image preprocessing, LPRNet directly performs license plate localization and character recognition from images, offering both efficiency and strong generalization capability.

Loffe [12] recommends the extensive use of batch normalization (BN) layers in neural networks while minimizing the use of dropout layers, as BN typically provides stronger generalization and more stable training behavior compared to dropout. In this study, the dropout layers in LPRNet are replaced with BN layers. To further address the issue of insufficient feature information fusion during LPRNet training, an Efficient Multi-scale Attention (EMA) module is incorporated to enhance the network's capacity for extracting and learning license plate image features. Specifically, the EMA mechanism is added after the second Small Basic Block of LPRNet, strengthening the network's ability to capture character-specific features and resulting in the improved N_E_LPRNet architecture, as illustrated in Figure 6.

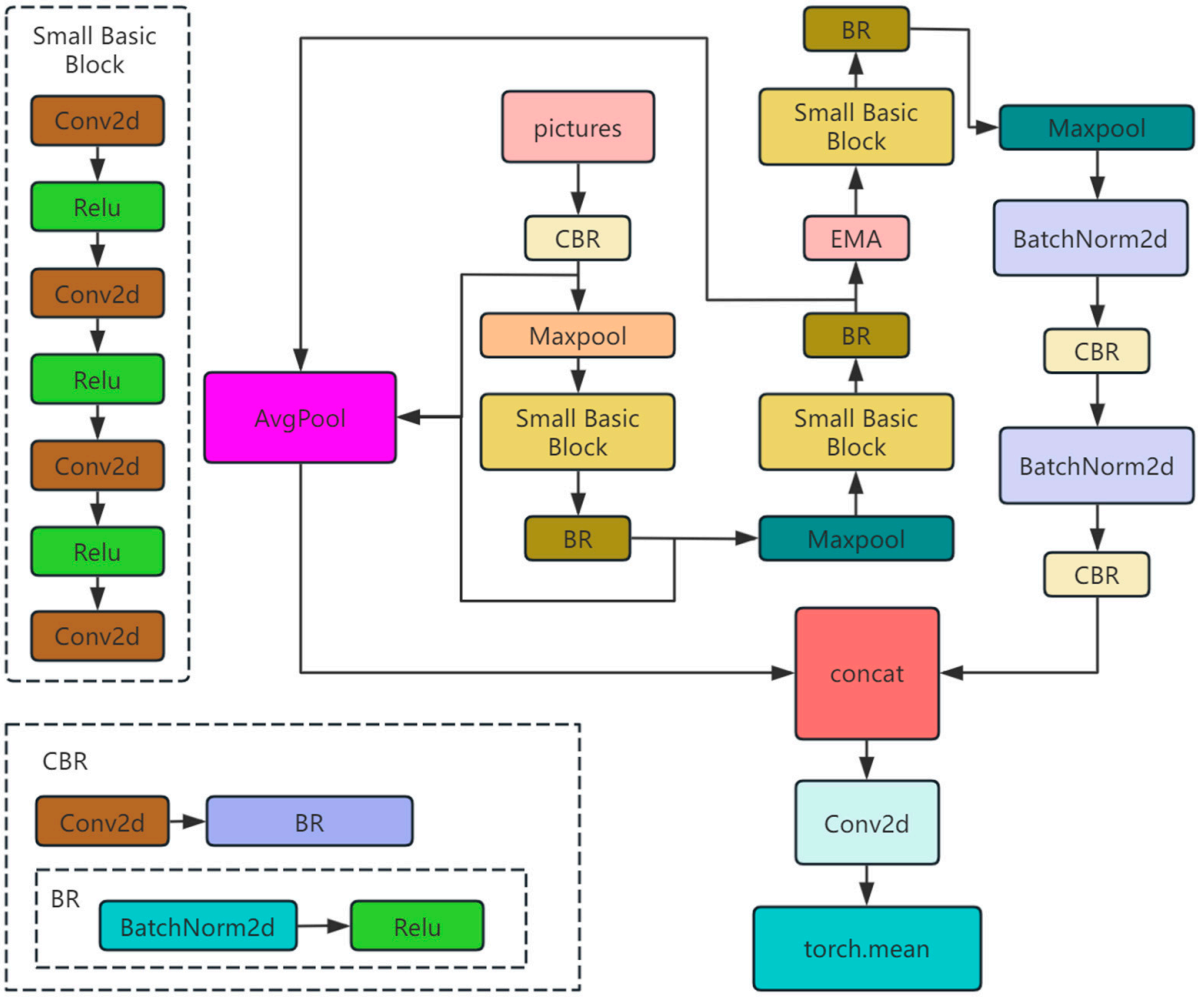

Figure 6. N_E_LPRNet network structure.

3. Experiment

3.1. Establishment and Processing of the Dataset

The datasets used for license plate location and license plate recognition in this experiment are not the same. For the license plate location part, we constructed the dataset by combining a portion of the public Chinese license plate dataset CPDD with some other license plate images from other provinces that were sorted in the CCPD format. The CCPD license plate dataset is an open-source and free Chinese urban license plate recognition dataset established by a team from the University of Science and Technology of China. Since in the CCPD dataset, each image only contains one license plate, and the license plate province is mainly Anhui, to prevent the problem of excessive samples of "Anhui" license plates in the dataset causing poor model performance, we selected a portion of the license plates from CCPD (including some car photos with tilted license plates) and a portion of the license plate images from other provinces sorted in the CCPD format to construct a dataset with a total of 13,000 images. The dataset was randomly divided into training set and validation set at a ratio of 7:3. For the dataset of the license plate location network, it was trained and tested using this self-built dataset. For the dataset of the license plate recognition network, it was composed of images obtained by cutting the images after the four-point positioning of the license plate and applying perspective transformation, which were clearer license plate images generated from the own dataset, totaling 13,000 images. It was divided into training set and validation set at a ratio of 7:3.

3.2. Experimental Environment

The experimental platform uses Python 3.9 version and is compiled under the PyTorch 2020 1.3 deep learning framework. It uses an NVIDIA 4060 graphics card with 8 GB of memory and CUDA version 12.1. For the YOLOv8n model, the hyperparameters used by the improved and original algorithms are the same, with a learning rate of 0.01, 100 epochs, and a batch size of 32. For the LPRNet model, the parameters used by the improved and original algorithms are all the same.

3.3. Evaluation Indicators

For the license plate detection task, this experiment measures the model performance based on three evaluation metrics: mean Average Precision (mAP), parameters, and floating-point operation count (GFLOPs) [9]. For the license plate recognition task, the performance of the model is evaluated based on three metrics: the number of parameters (Parameters), the number of floating-point operations (GFLOPs), and the accuracy rate of the model on the validation set.

Precision refers to the proportion of all positive samples that are actually positive in the model's predictions. The formula for average precision is as follows:

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \int_0^1 P(R)dR \tag{9}$$

$$mAP = \frac{\sum_{i=1}^{i=n} AP_i}{n} \tag{10}$$

In the above formula, P represents the accuracy of the model. The higher the P value, the stronger the model's ability to predict positive samples. TP represents true positive

examples, *FP* represents false positive examples, and *FN* represents false negative examples. *R* is the recall rate, and *mAP* is the average precision mean, which measures the average precision of the model across all target categories.

The parameter quantity refers to the total number of all learnable parameters in the model. During the training process, these parameters are automatically updated through algorithms. The larger the parameter quantity, the more complex the theoretical function space that the model can represent, thereby capturing more complex features.

The 10 billion floating-point operations per second (GFLOPs) indicates the computational resources required by the model during inference and serves as a key metric for evaluating model efficiency and performance [9]. Lower GFLOPs correspond to reduced computational complexity and faster inference speed, making the model more suitable for deployment in scenarios with limited computational power and energy resources. This study primarily focuses on model compactness and computational efficiency, requiring a trade-off between the model's representational capacity and its computational demands.

3.4. Comparative Experiment

Regarding the license plate detection part, we compared the improved YOLOv8 with several models, including YOLOv5s, YOLOv5n, YOLOv8s, YOLOv8n, YOLOv11s, YOLOv11n, DETR-L, and DETR-RESNRT50. The comparison results of various models are shown in Table 1.

Table 1. The results of the license plate detection comparison experiment.

Model	Param (10^6)	GFLOPs	Box: Map@ 0.5–95	Size MB
YOLOv5s	7.32	16.7	0.797	14.73
YOLOv5n	1.92	4.5	0.736	3.86
YOLOv8s	11.64	30.4	0.825	22.1
YOLOv8n	3.29	9.3	0.778	6.09
YOLO11s	9.71	22.5	0.834	18.8
YOLO11n	2.66	6.7	0.783	5.32
DETR-L	32.8	108	0.844	63.0
DETR-RESNRT50	42.7	130.5	0.851	82.0
GCE_B_YOLO	2.25	6.7	0.769	4.17

From Table 1, it can be seen that our model is not as accurate as YOLOv5s, YOLOv8s, YOLO11s, DETR-L and DETR-RESNRT50. As can be seen from the table, the model using DETR-RESNRT50 has the highest accuracy, and its generated model size is also the largest. Our model has a certain level of accuracy while its model size is only 5% of that of the others. The smaller the model size, the more suitable it is for deployment in some embedded devices with lower performance than a computer. The number of parameters in our model is much less than those of these models, which enables our model to be applied on platforms with limited performance. Although the number of parameters in our model is slightly more than YOLOv5n, and less than YOLOv8n, YOLO11n, our model's accuracy is much higher than YOLOv5n, and is basically the same as the accuracy of YOLOv8n and YOLO11n. Therefore, it can be concluded that our model has the characteristics of both high accuracy and light weight. Compared with the models in the table, our improved model has achieved a balance between model accuracy and model size.

For the license plate recognition part, our model was compared with other license plate recognition models such as CRNN + CTC, and LPRNet. The comparison results of various models are shown in Table 2.

From Table 2, it can be seen that the license plate recognition accuracy of CRNN + CTC is much higher than that of the initial LPRNet. However, their model sizes are also

much larger than those of the initial LPRNet. Compared with the improved model and CRNN + CTC, the improved model has a very small size, being only 15% of the original model's size, and has a higher accuracy rate. Compared with the original LPRNet, the improved model has a significant improvement in accuracy while maintaining almost the same number of parameters and model size. The improved model has achieved good results in balancing accuracy and size.

Table 2. The results of the license plate recognition comparison experiment.

Model	Param (10^6)	GFLOPs	Accuracy Rate %	Size MB
CRNN + CTC	3.40	1.98	98.47	11.6
LPRNet	0.44	0.14	94.36	1.73
N_E_LPRNet	0.44	0.16	96.61	1.74

3.5. Ablation Experiment

To analyze the impact of the GCE module and the feature fusion network BIFPN we designed on the model size and the model's localization effect, we conducted ablation experiments and compared the results. The experimental results of the improved YOLOv8n and the original YOLOv8n are shown in Table 3. Here, "A" indicates the model with "GCE" replaced, and "B" indicates the model with the feature fusion network BIFPN replaced.

Table 3. The ablation experiment results of license plate detection.

Model	Param (10^6)	GFLOPs	Box: Map@ 0.5–95	Size MB
YOLOv8n	3.29	9.3	0.778	6.09
YOLOv8n + A	2.23	6.6	0.752	4.14
YOLOv8n + B	3.31	9.3	0.787	6.12
YOLOv8n + A + B	2.25	6.7	0.769	4.17

This study conducted ablation experiments to evaluate the effectiveness of different optimization strategies. As shown in Table 3, the GCE structure substantially reduced the number of parameters in the YOLOv8n model by 32.2% and decreased the model size by 1.95 MB; however, it resulted in a 2.6-percentage-point drop in the model's mAP, slightly affecting accuracy. The BiFPN structure led to a modest increase in mAP, demonstrating its effectiveness in improving small target detection, such as license plates. Finally, the combination of the GCE and BiFPN structures in the improved YOLOv8n model reduced the number of parameters by 31.6% while causing only a minimal decrease in mAP. Compared to the original model, which experienced approximately a 1-percentage-point drop in mAP while achieving a 1.95 MB reduction in model size, this approach successfully achieved a compact and computation-efficient design without significantly compromising detection accuracy.

To analyze the impact of the dropout structure we replaced in the LPRNet network and the EMA module we introduced on the model, we conducted ablation experiments and compared the results. The experimental results of the improved LPRNet and the original LPRNet are shown in Table 4. Here, "A" indicates that the "dropout" model was replaced in the network, and "B" indicates that the EMA mechanism was added to the network.

As shown in the table, the improved LPRNet exhibits negligible differences in terms of parameters and model size compared to the original algorithm, while still satisfying the requirements for compactness and computational efficiency.

Table 4. The ablation experiment results of license plate recognition.

Model	Param	GFLOPs	Size/MB
LPRNet	446,976	0.1478	1.73
LPRNet + A	447,648	0.1633	1.73
LPRNet + B	447,616	0.1482	1.73
LPRNet + A + B	448,288	0.1637	1.74

Taking into account the accuracy and stability of the model recognition, we conducted experiments on the improved LPRNet model. Using the self-built LPRNet dataset, we conducted 30 tests on four algorithms (LPRNet, LPRNet + bn, LPRNet + EMA, and LPRNet + bn + EMA). The results obtained are shown in Table 5.

Table 5. The accuracy of the character recognition algorithm.

Model	Maximum Value/ %	Minimum Value/ %	Average Value/ %	Mid-Value/ %	Variance
LPRNet	95.62	93.73	94.41	94.36	2.83×10^{-6}
LPRNet + A	95.36	94.83	95.09	95.12	1.92×10^{-6}
LPRNet + B	97.03	94.88	95.92	95.82	2.91×10^{-6}
LPRNet + A + B	96.97	96.31	96.56	96.61	2.08×10^{-6}

From the above table, it can be seen that the LPRNet algorithm with batch normalization structure has a more stable recognition rate compared to the original LPRNet algorithm with dropout structure. Meanwhile, the LPRNet algorithm with EMA has a recognition rate increase of more than one percent compared to the original LPRNet algorithm. This indicates that our improvements have indeed enhanced the performance of the model.

In addition, considering the recognition speed of the model, we conducted experiments on the improved LPRNet model. We tested the four algorithms of LPRNet, LPRNet + EMA, LPRNet + bn, and LPRNet + bn + EMA using our self-built LPRNet dataset for 30 times. The recognition speeds obtained are shown in Table 6.

Table 6. The recognition speed of the character recognition algorithm.

Model	Maximum Value/ ms	Minimum Value/ ms	Average Value/ ms	Mid-Value/ ms
LPRNet	65	32	48	50
LPRNet + A	63	31	46	48
LPRNet + B	68	34	49	51
LPRNet +A + B	66	30	50	53

From the above table, it can be seen that the recognition speeds of these four algorithms are close, all reaching a response time of 50 ms for the recognition results. The average recognition speed of the LPRNet + bn + EMA algorithm is slightly higher than the others. This slight difference is mainly due to the fact that its network structure is more complex than the original one, so it takes a little more time to process, and here we explain that our improved model has almost no burden on the equipment and does not affect the recognition speed.

We conducted experiments by comparing the improved license plate recognition system with the unimproved one. We provided the same license plate images as input to both systems. The model size of the system and its recognition speed and accuracy rate for the images were used as reference criteria. The experimental data are shown in Table 7.

Table 7. The recognition speed of the character recognition algorithm.

Model	Accuracy Rate %	Size/ MB	Value/ ms
YOLOv8n + LPRNet	0.93	7.82	251
GCE_B_YOLO + LPRNet	0.96	5.9	230
YOLOv8n + N_E_LPRNet	0.94	7.83	275
GCE_B_YOLO + N_E_LPRNet	0.97	5.91	235

From the table, it can be seen that the volume of our improved YOLO and LPRNet models has decreased by 25% compared to the original ones. Moreover, our model has also achieved certain improvements in accuracy. Based on the overall performance of the experiments, it can be concluded that our model has achieved good results in realizing license plate recognition with low computational costs.

3.6. Some Visualized Experimental Results

The following figures illustrate the system in operation during our experiments. Figures 7 and 8 show that the license plate detection and recognition models are functioning correctly, while Figure 9 demonstrates the overall system operating normally. From this, it can be seen that our improved model has a very good recognition effect.



Figure 7. The license plate detection part.



Figure 8. The license plate recognition part.

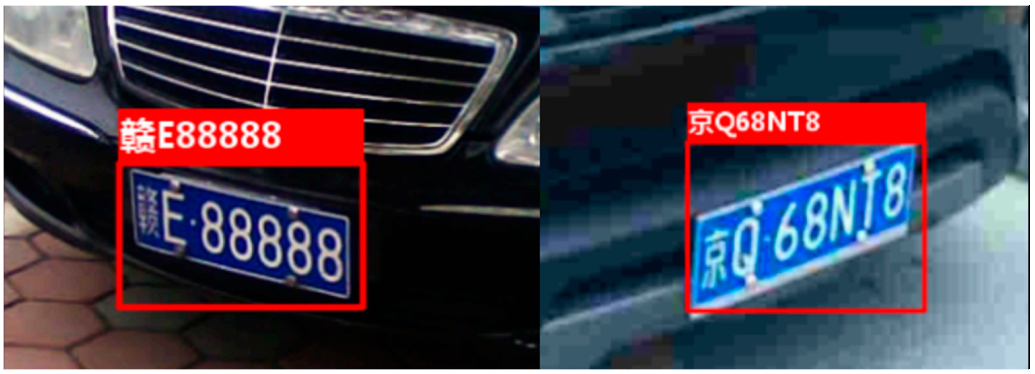

Figure 9. License plate recognition system.

4. Conclusions

This paper addresses the challenge of deploying license plate recognition algorithms on resource-constrained devices and proposes an improved compact YOLOv8n + LPRNet framework to facilitate such deployment. First, the Ghost convolution is adopted to construct a novel GCE module, which is integrated into YOLOv8n. Concurrently, a weighted bidirectional feature pyramid network (BiFPN) is incorporated into the neck to enhance multi-scale feature fusion. Second, the EMA module and batch normalization are introduced to optimize the LPRNet network. The improved detection and recognition models are combined and evaluated through experiments. Experimental results demonstrate that the enhanced license plate detection model achieves accuracy comparable to the original model while significantly reducing the number of parameters and decreasing the model file size by approximately 32%. The optimized license plate recognition model exhibits higher and more stable recognition accuracy. When combined with the detection model, the total size is only 5.91 MB, offering clear advantages in compactness and computational efficiency, making it suitable for deployment on resource-constrained devices.

Although the proposed model achieves a favorable balance between accuracy and efficiency, there remains room for improvement. The primary dataset mainly consists of license plates captured in standard conditions; therefore, detection and character recognition performance may degrade under challenging scenarios such as heavy smog or extremely bright or dim lighting. Future work may involve designing image preprocessing techniques to handle license plates under diverse environmental conditions, thereby improving recognition accuracy and enhancing the model's adaptability.

Author Contributions: Conceptualization, S.Y.; methodology, X.Z.; software, X.Z.; validation, S.Y. and X.Z.; formal analysis, S.Y. and X.Z.; resources, X.Z.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, S.Y.; visualization, X.Z.; supervision, S.Y.; project administration, S.Y.; funding acquisition, S.Y.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Xu, G.Z.; Kuang, W.; Li, X.W.; Wan, Q.B.; Shi, Y.T.; Lei, B.J. License Plate Location Based on YOLOv3 and Vertex Offset Estimation. *Comput. Aided Des. Comput. Graph.* **2021**, *33*, 569–579. [CrossRef]
2. He, Z.L.; Xiao, Z.J.; Yan, Z.G. Lightweight Convolutional Network for License Plate Recognition. *Qilu Univ. Technol.* **2020**, *34*, 35–41.
3. Rao, W.J.; Gu, Y.H.; Zhu, T.T.; Huang, Y.T. Intelligent License Plate Recognition Method in Complex Environment. *Chongqing Univ. Technol. Nat. Sci.* **2021**, *35*, 119–127.
4. Guo, S.X.; Zhang, H. Embedded Device Realizes Low Light License Plate Image Recognition. *Comput. Digit. Eng.* **2022**, *50*, 881–886.
5. Wu, H.W. Research on Vehicle License Plate Detection and Recognition System Based on Deep Learning. Master's Thesis, Dalian University of Technology, Dalian, China, 2021.
6. Cao, J.A.; Yang, W.M.; Luo, Y.T.; Pan, N.Y.; Zhang, W. Design of license plate recognition algorithm based on deep learning. *Mod. Electron. Technol.* **2025**, *48*, 135–139.
7. Andrew, G.; Zhu, M.L. Efficient convolutional neural networks for mobile vision applications. *Mobilenets* **2017**, *10*, 151.
8. Zhang, J.H.; Yang, Z.Y.; Xia, L.H.; Liang, Z.W. Lightweight Road Traffic Sign Detection Method with Attention Mechanism. *Electron. Meas. Technol.* **2023**, *46*, 85–92.
9. Huang, C.Q.; Xu, H.Y.; Zhang, X.L.; Zhu, X.Z. BGR-YOLO: An Improved Object Detection Algorithm Under Traffic Scenarios Based on YOLOv8. *Comput. Eng. Sci.* **2025**, *15*, 1–13.

10. Behera, S.K.; Dash, S.P.; Amat, R.; Sethy, P.K. Wafer defect identification with optimal hyper-parameter tuning of support vector machine using the deep feature of ResNet 101. *Int. J. Syst. Assur. Eng. Manag.* **2024**, *15*, 1294–1304. [CrossRef]
11. Ullah, A.; Elahi, H.; Sun, Z.; Khatoon, A.; Ahmad, I. Comparative analysis of AlexNet, ResNet18 and SqueezeNet with diverse modification and arduous implementation. *Arab. J. Sci. Eng.* **2022**, *47*, 2397–2417. [CrossRef]
12. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML. Pmlr.* **2015**, *37*, 448–456.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.