

RESEARCH ARTICLE

License Plate Recognition System Based on Improved YOLOv5 and GRU

HENGLIANG SHI¹ AND DONGNAN ZHAO^{ID}²¹Automobile and Rail Transit School, Luoyang Polytechnic, Yibin, Luoyang 471023, China²Information Engineering School, Henan University of Science and Technology, Luolong, Luoyang 471000, China

Corresponding author: Dongnan Zhao (Zhao_dn2020@163.com)

This work was supported in part by the Key Scientific Research Projects of Henan Province Education Institution under Grant 22B120002.

ABSTRACT Aiming at the problem that the traditional license plate recognition method lacking of accuracy and speed, an end-to-end deep learning model for license plate location and recognition in natural scenarios was proposed. First, we added an improved channel attention mechanism to the down-sampling process of the You only look once (YOLOv5). Additionally, a location information is added in the ones to minimize the information loss from sampling, which can improve the feature extraction ability of the model. Then we reduce the number of parameters on the input side and set only one class in the YOLO layer, which improves the efficiency and accuracy of the detector for locating license plates. Finally, Gated recurrent units (GRU) + Connectionist temporal classification (CTC) was used to build the recognition network to complete the character segmentation-free recognition task of the license plate, significantly shortened the training time and improved the convergence speed and recognition accuracy of the network. The experimental results show that the average recognition precision of the license plate recognition model proposed in this paper reaches 98.98%, which is significantly better than the traditional recognition algorithm, and the recognition effect is good in complex environment with good stability and robustness.

INDEX TERMS Deep learning, target detection, license plate recognition, YOLOv5, GRU.

I. INTRODUCTION

The license plate is an important information carrier of the vehicle, providing a unique identity mark for the vehicle. License plate recognition is a key link in building intelligent transportation, which can play an important role in traffic calming, vehicle tracking, unmanned parking lots, and automatic highway toll collection. The current stage of license plate recognition technology is generally divided into three stages: detection, segmentation and recognition. Such schemes have complex processes, low efficiency, and are easily affected by uneven lighting and noise, with poor robustness.

Although license plate recognition technology has been widely used in real life, it is more often used in fixed scenarios and environments, and the precision and robustness of existing recognition technology can hardly meet the needs of realizing applications in complex conditions and real-time

The associate editor coordinating the review of this manuscript and approving it for publication was Mouloud Denai ^{ID}.

scenarios. In recent years, with the rapid development of computer hardware, neural network models based on deep learning have become the best tools to solve complex computer vision problems [1], [2], [3]. Convolutional Neural Network (CNN) is one of the best deep learning techniques for target detection and recognition tasks, and the most popular algorithm in CNN-based target detection is YOLO, proposed by Redmon in 2015 [4]. It creatively combines the two stages of candidate area and target recognition into one, and only one forward operation is needed to complete the target detection, which greatly reduces the image processing time and makes the model very efficient, many researchers have done a lot of secondary development work on the YOLO family of model detectors [5], [6]. On the one hand, there is cooperation with other methods, such as use YOLOv3 to extract and classify underwater objects and combine it with a deep learning method based on (Long Short Term Memory) LSTM to determine the location of the underwater objects [7]. On the other hand, the YOLO backbone structure is optimized [8], [9], such as replaced the output layer with deformable

convolution to improve the detection speed in the backbone network CSPDarknet53_dcn(P) of YOLOv4. And a new feature fusion module was redesigned to improve the detection accuracy of small objects using multiple scale detection layers [10]. The latest version of YOLOv5 is highly precise, fast, and generates detection weight files of only 10-120 MB using Pytorch framework, which means that high-precision license plate detection models have become much easier to use on embedded devices [11], [12]. Such as constructed an auxiliary domain (S-DAYOLO) based on YOLOv5s, in which the synthetic image uses the latest method to convert the source image to a similar image in the target domain, which is a good solution to the problem of object detection performance degradation after different domain conversions. It will be embedded in electronic components for use in autonomous driving assistance systems [13]. Meanwhile, recurrent neural networks(RNNs) play an indispensable role in the field of natural language processing by processing characters directly as input sequences, eliminating the processing of character segmentation and allowing end-to-end license plate recognition models to become mainstream [14], [15]. In model optimization, attention mechanisms enable models to know “where the places of interest are” and are widely used to improve the performance of neural network models [16], [17]. Among them, SE(Squeeze and Excitation) is the most popular attention mechanism because of its low cost and high gain by establishing channel correlation through 2D global pooling [18]. However, the SE attention mechanism only collects information about the relationship between channels and does not focus on the corresponding location information, this information is extremely critical for the acquisition of target structures in detection and recognition tasks [19].

The main contributions of this paper are as follows:

- The proposed lightweight deep learning model requires only one forward computation process to complete the end-to-end detection and recognition of license plates;
- The YOLOv5 algorithm is improved to extend a novel attention mechanism in the down-sampling process of the Neck structure, this work improves the efficiency and accuracy of license plate location;
- Upon using the improved YOLOv5, we modified the feature parameters of the prediction part of the classifier to increase the accuracy of the model while reducing the training time;
- We use GRU + CTC recognition network to complete the recognition of positioned license plates. The model does not require pre-segmentation of license plate characters, and the automatic extraction of characters is done by deep neural networks after self-learning.

To demonstrate that the proposed method in this paper is more effective compared with previous license plate recognition algorithms, we conducted extensive experiments on the CCPD dataset. With the same training and test sets, our recognition algorithm improves the recognition precision by 0.44%. In terms of algorithm operational efficiency, we also observe that although an improved attention mechanism is

incorporated, a more efficient recognition network structure ensures that the method meets the requirements of practical engineering.

Following the overview in the first section of this paper, We present the work related to license plate detection and recognition algorithms in Section I. The basic framework of the YOLOv5-LSE method and the optimization process are presented in Section II. To verify the effectiveness of the method, we show its experimental results on the dataset in Section III and perform a detailed comparative analysis with other cutting-edge recognition algorithms. Subsequently, the Improved algorithm is discussed in section IV.

II. RELATED WORKS

In the license plate positioning stage, the traditional License plate localization methods based on a priori information are generally classified as color texture [20], [21], shape regression [22], [23], and edge detection [24], [25]. The color of the license plate is usually blue, yellow, white and green, with high color contrast and fixed shape, so the color and shape features are widely used. For example, Tian et al. [26] used a color difference model to obtain a binarized image to select the target region, then used the Adaboost algorithm to train the above features along with other features to obtain a classifier, and finally used the classifier to precisely locate the license plate. SalauAO et al. [27] used the license plate aspect ratio geometric information as a threshold for foreground extraction to implement the GrabCut algorithm for automatic license plate localization. This approach has limitations because the aspect ratios of the license plate is different from place to place. The feature extraction of traditional localization methods relies on manual design, which is not well suited to the diversity of images. Therefore, traditional methods of license plate detection are inefficient and have poor accuracy. In recent years, target detection methods based on deep learning have developed rapidly, and the algorithms are mainly divided into two categories. In one category, a part of the candidate region is first generated by the algorithm, and then the candidate region is classified and positioned again [28], [29]. For example, Naaman Omar et al. [30] used a deep semantic segmentation network to classify license plate images into digital regions, city regions, and country regions. Slimani et al. [31] based on wavelet transform for license plate detection, followed by validation of potential regions using CNN classifier. Another category is end-to-end detection algorithms, which directly get the location coordinates and class probability of the target, typical algorithms are SSD [32], YOLO [33], [34], [35]. The former has a lower recognition speed and the latter is slightly less accurate.

In the license plate recognition stage, the traditional recognition algorithm usually performs the operation of segmenting the license plate characters one by one first, and then uses optical character recognition (OCR) technology to recognize each character [36], [37]. Nahlah M [38] uses the honeybee algorithm to complete the segmentation of the license plate

characters and then uses a support vector machine (SVM) to recognize the license plate characters. Experimental results show that the method has a good license plate recognition effect, but it has poor recognition efficiency. The current popular recognition algorithms discard the character segmentation process and more often fuse license plate detection with recognition [39], [40]. Such as Omar et al. [30] proposed the concept of LPR-CNN, which consists of two convolutional neural networks to form a license plate and character detection system. This method is trained in an end-to-end manner and does not require pre-formed character segmentation. Experimental results show that the method is effective for license plate recognition of real vehicle images. In addition, in practical applications, many license plate recognition algorithms can only achieve good recognition under specific conditions [41]. This condition includes good weather conditions, adequate lighting, fixed scenes, and facilities. License plate recognition in complex environments remains difficult, with challenges such as poor lighting at night, rain and snow, dumped, obscured or blurred license plates [42]. In traditional license plate recognition, the two modules, location and recognition, are usually divided into two separate tasks, and use more complex algorithms to solve these challenges. However, the neural network model based on deep learning connects the two problems well. Therefore, this paper proposes an end-to-end license plate detection and recognition method based on deep learning that optimizes the efficiency and accuracy of recognition.

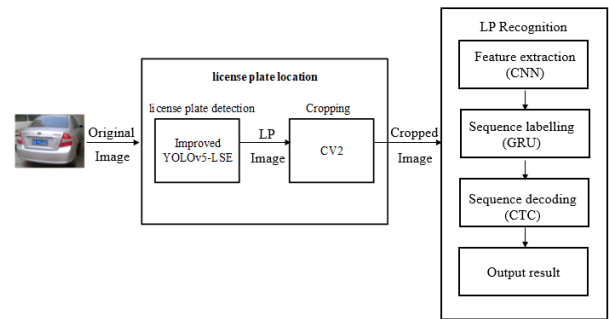


FIGURE 1. LPR model framework.

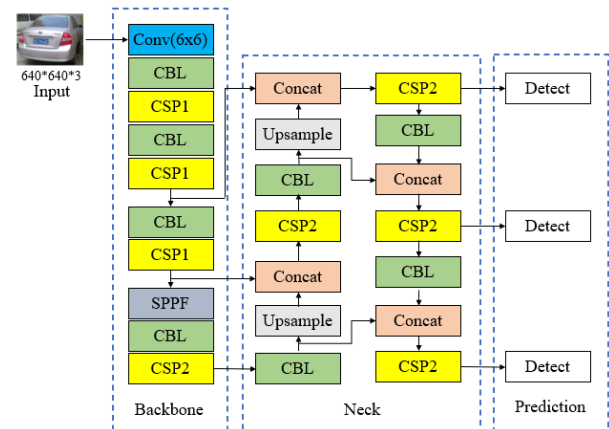


FIGURE 2. YOLOv5 network structure.

III. METHODOLOGY

A. MODEL FRAMEWORK

The overall network model framework of the license plate recognition system is shown in Fig.1, which consists of two parts: license plate positioning and license plate recognition, and synthesizes the two parts into an overall network model through the data interface. First, in the license plate localization module, we use the improved YOLOv5 model to perform the detection and cropping work on the license plates in the images. After that, in the recognition network, we use GRU to complete the work of sequence labeling and decoding. The GRU output matrix and the corresponding ground-truth(GT) text will be input into the CTC loss function, and the output recognized license plate results will be obtained by calculating the loss function values of each data point.

B. YOLOv5

YOLO is one of the most famous object detection algorithms because of its high efficiency, high precision and light weight. YOLOv5 is the latest generation of the YOLO family of detection networks introduced by Glenn Jocher in May 2020 using Pytorch framework. There are four versions of the YOLOv5 network model: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5 network has the smallest depth and feature map width in the YOLOv5 series, and the next 3 are deepened and widened on its basis.

As shown in Fig.2, the network structure of YOLOv5 consists of four parts: input, backbone, neck and prediction. YOLOv5 has been iterated to V6.1 version, and we will also use the latest version of the network structure in this paper, on which the related model introduction and improvements are based.

Input: The input is the stage of image pre-processing for the input image. Preprocessing includes data enhancement, adaptive image scaling and anchor frame calculation. YOLOv5 uses the Mosaic data enhancement method to stitch four images into a new photo by random layout, cropping and scaling, which greatly enriches the detection. And the data of four images can be calculated directly in the calculation of batch normalization, which speeds up the training efficiency. YOLOv5 has embedded the anchor frame calculation into the training, outputting the predicted frame on the initial anchor frame, and later comparing it with Ground-truth to calculate the Loss, thus continuously updating the anchor frame size and adaptively calculating the optimal anchor frame value.

Backbone: The Backbone mainly consists of Focus structure and CSP structure. However, after the latest version V6.0 of YOLOv5, the Focus module is replaced with a 6×6 sized convolutional layer. The two are theoretically equivalent, but for some existing GPU devices and corresponding optimization algorithms, the 6×6 convolutional

layer is more efficient. The CSP structure enhances the learning ability of the model and speeds up network inference.

Neck: The Neck includes FPN and PAN structures. YOLOv5 adds PAN to the FPN structure to make the combined structure better for the fusion and extraction of features at different levels. Version V6.1 also replaces SPP with SPPF, which is designed to convert arbitrarily sized feature maps into fixed-size feature vectors more quickly.

Prediction: Prediction completes the output of the target detection results. YOLOv5 forms a new loss function CIOU based on the IOU loss function by considering the distance information of the center point of the bounding frame, and IOU refers to the intersection ratio of the predicted frame and the real frame. DIOU_nms is also used in this process instead of the traditional NMS operation, with the aim of suppressing redundant frames better, thus further improving the detection accuracy of the algorithm.

C. DETECTION ALGORITHM OPTIMIZATION

In general, the license plate takes up a relatively small part of the image, and there are two problems with license plate detection as follows:

- The information presented by the pixels in the detection area where the license plate is located is limited, which can easily lead to poor detection of the license plate by the target detection algorithm;
- In the model training phase, the labeling of small objects is prone to bias, and the detection results will be greatly affected when the targets are small and the number of classes is large.

To solve the above two problems, we have improved the YOLOv5 algorithm. We improve the feature extraction ability of the model by adding an attention mechanism to improve the detection effect of the model on small targets. At the same time, we use a single class in the model, which greatly reduces the number of parameters, makes the model less likely to fall into category confusion, and reduces the impact of labeling on detection results. The two improved parts will be explained in detail in the following two subsections.

1) NOVEL ATTENTION MECHANISMS

The neck structure of YOLOv5 is the FPN + PAN model. FPN is a top-down feature pyramid that passes the higher-level semantic features down through up-sampling and convolution. But FPN only enhances the semantic information and does not pass on the localization information. The PAN structure compensates nicely for this by adding a bottom-up feature pyramid after the FPN. PAN performs a down-sampling operation on the bottom layer of the FPN, its upper layer will be subjected to 3×3 convolution operation, then it will be connected laterally with the bottom layer after down-sampling, and the two will be added together, and finally 3×3 convolution will be performed again to fuse their features. This process operates iteratively from the bottom of the FPN up to form a new feature pyramid that contains both semantic and localization information. PAN

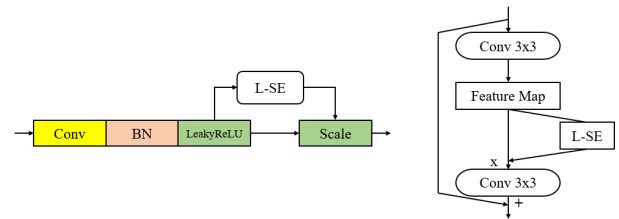


FIGURE 3. Attention mechanism.

uses 8 times, 16 times and 32 times down-sampling and convolution for three different sizes of images to complete the feature extraction and transfer. In this process, a large amount of position information is lost, making the model less accurate in detecting small targets.

In recent years, attention mechanisms have been widely used to improve the performance of modern deep neural networks. We introduce a new attention mechanism in order to solve the problem presented in the previous section [43]. The new attention mechanism is an improvement on the SE block (Squeeze-and-Excitation). The SE block focuses on the relationship between channels and allows the model to automatically identify the importance of different channel features, but ignores the location information. Location information is very important in the visual task of capturing target structure [18], and PAN will generate a large amount of channel and location information, we embed location information into channel attention to form a new attention mechanism L-SE. Its implementation in YOLOv5 is shown in Fig.3. It will be added to the down-sampling process in the PAN structure [44].

We will give a brief description of the SE module to better explain L-SE. The standard convolution itself is difficult to obtain channel relationships, but this channel relationship information is significant for the final classification decision of the model. The SE module does this job well by making the model focus more on the most informative channel features and suppressing the unimportant channel features to achieve better feature extraction. It works as follows: Firstly, the Squeeze is performed on the feature map obtained by convolution to get the global features of channels, then the Excitation is performed on the global features to learn the relationship between each channel and get the weights of different channels, and finally the final features are obtained by multiplying with the original feature map. Given the input U , the Squeeze equation for the c -th channel is as follows:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j), \quad z \in R^c \quad (1)$$

where Z is the global feature of the c -th channel, it is obtained by encoding the entire space on the channel with features. u_c come from the convolutional layers in the PAN structure- and their convolutional kernel size is fixed, so they can be considered as a collection of local descriptors. H denotes the height of the feature map and W is the width of the feature

map. F_{sq} denotes the global averaging of the squeezed work of the feature map in the set.

After we get the global description of the features by squeezing, we next need to get the relationship between the pipes through the excitation operation. The traditional SE module uses a gating mechanism in the form of sigmoid:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \text{ReLU}(W_1 z)) \quad (2)$$

where F_{ex} denotes the excitation operation, $W_1 \in R^{\frac{c}{r} \times c}$, $W_2 \in R^{c \times \frac{c}{r}}$, they represent two linear transformations, and the weights of each channel are obtained by learning them. The *sigma* is a nonlinear activation function that normalizes to obtain the importance of the channel, with 0 being unimportant and 1 being important.

The optimization of L-SE lies in the addition of location information. Equation (1) only squeezes the global spatial information and does not preserve the location information. equation (1) only squeezes the global spatial information and does not preserve the location information. We factorize equation (1) into equation (2) parts ($V, 1$) and ($L, 1$), representing two spatial ranges of pooling kernels, which operate in pairs of 1D feature encoding for each channel in different directions, ($V, 1$) being vertical and ($L, 1$) being horizontal. The output of the c -th channel with height v in the vertical direction is:

$$z_c^v(v) = \frac{1}{L} \sum_{i=1}^L u_c(v, j) \quad (3)$$

In the same way, the output of the c -th channel at width l in the horizontal direction can be written as:

$$z_c^l(l) = \frac{1}{V} \sum_{i=1}^v u_c(i, l) \quad (4)$$

The above two formulations will collect features along two spatial directions, horizontal and vertical, and will eventually generate a pair of perceptual feature maps in the corresponding directions. It differs from the squeeze operation in that L-SE can learn the relational weights between individual channels in one spatial direction while collecting precise position information in another spatial direction. This approach helps YOLOv5 to locate the target of interest more precisely [45].

In order to make full use of the collected location information, we propose a new method for calculating weights. Create a shared 1×1 convolutional transform function F into which a subset of the aggregated features generated by equation (3) and equation (4) will be fed, the excitation operation yields:

$$f = \sigma\left(F\left(z^v, z^l\right)\right) \quad (5)$$

where (z^v, z^l) denotes the tandem operation of aggregated feature subsets and σ a nonlinear activation function. where $f \in R^{c/r \times (V+L)}$ denotes an intermediate feature map with information encoded in two different directions. Similarly, we split

f into two separate tensors along the direction $f \in R^{c/r \times (V+L)}$ and $f^v \in R^{c/r \times V}$. We use a bottleneck structure with two fully connected layers here to reduce model complexity as well as to improve generalization capabilities. In the first FC layer, r is a dimensionality reduction coefficient, which plays the role of dimensionality reduction. In the second FC layer, f^v and f^l are transformed to have the same dimension as the input U by two 1×1 convolutional transforms F_v and F_l , which are independent of each other. Yielding:

$$w^v = \sigma\left(F_v\left(f^v\right)\right) \quad (6)$$

$$w^l = \sigma\left(F_l\left(f^l\right)\right) \quad (7)$$

where σ is the sigmoid function, w^v and w^l are used as attention weights for the different channels. Finally, the Scale operation is performed, which means that the obtained weights are multiplied with the original feature map to get the final features. The output of L-SE block X is:

$$x_c(i, j) = u_c(i, j) \times w_c^v(i) \times w_c^l(i) \quad (8)$$

As mentioned above, the L-SE block not only considers the importance of the different channels, but also focuses on the coded location information. We apply the attention of two different directions simultaneously to the input tensor, and the resulting attentional map can determine whether the corresponding direction is storing the target of interest. We can also adjust the attention during this encoding process to make the localization of the interest target location more accurate and thus improve the target detection ability of the model.

2) PARAMETRIC OPTIMIZATION

YOLOv5 was originally implemented on the COCO2017 dataset, with 80 classes present in the original classifier (people, motorcycles, fire hydrants, elephants, umbrellas, etc.). In YOLOv5, each bounding box is represented by five predicted values, and RGB has three channels, so the number of parameters for predicting only the bounding boxes is $3 \times (5 + 80) = 255$. This number of parameters is too large, which will reduce the prediction efficiency of the model while increasing the probability of class errors. In this paper, our second contribution is to reduce the number of classes in the classifier. In the model where only a single class (license plate) is used, the number of parameters in the prediction bounding box will become $3 \times (5 + 1) = 18$. Such improvements make model detection much faster and less likely to fall into error confusion, thereby ensuring the accuracy of model detection.

D. RECOGNITION ALGORITHM

Character segmentation is an inseparable part of the traditional license plate recognition framework. The effect of character segmentation is highly susceptible to noise and complex environment. If the effect of character segmentation is not good, even if we have a high-performance recognizer, there will be false recognition and missed recognition. Many different image pre-processing methods are generally used in traditional recognition schemes to solve this problem, but

none of them has achieved better results. So we treat the characters in license plates as undivided sequences and use a deep learning model to solve the recognition problem.

1) GRU + CTC

In the field of end-to-end license plate recognition algorithms, the LSTM + CTC scheme is more widely used. However, in recent years, the emergence and development of GRU gives us more options. GRU is a very effective variant of LSTM network, which is simpler in structure, lighter in weight, and very effective in recognition. Therefore, in this paper, we choose the combination of GRU + CTC as the license plate recognition algorithm, and we also prove through experiments that this scheme improves the model training time, convergence speed, and recognition accuracy.

2) SEQUENCE SIGNATURE GENERATION

The process of license plate recognition based on GRU + CTC is shown in Fig.4. The recognition process of GRU + CTC based license plate recognition model is shown in Fig.4. In the first part of the license plate recognition model, we use a pre-trained 7-layer CNN model to extract the sequential feature representation from the cropped license plate image. A conv5 feature map of size $N \times C \times W \times H$ is obtained, after which a sliding window of 3×3 is made on conv5, and each point will be combined with the features of the surrounding area to obtain a feature vector of length $3 \times 3 \times C$, and a feature map of $N \times 9C \times W \times H$ is output. But the obtained features are spatial features, which only CNN can learn, and we will do Reshape operation to transform them to the size that GRU can learn.

3) SEQUENCE LABELLING

We will use GRU recursive processing for each layer of features in the obtained feature sequence. GRU allows prediction of past contextual information, and the network makes the sequence recognition operation more stable compared to processing each feature individually. GRU has only two gates, which combine the input gate and the forget gate in LSTM into one, called update gate, to select the memory information that can continue to be retained until the current moment. The other gate is the reset gate, which will control how much of the past information is forgotten. We first need to get the two gating states by the last transmitted down state h^{t-1} and the input x^t of the current node. The equation is:

$$z_t = \sigma \left(W_{xz}x^t + W_{hz}h^{t-1} + b_z \right) \tag{9}$$

$$r_t = \sigma \left(W_{xr}x^t + W_{hr}h^{t-1} + b_r \right) \tag{10}$$

where σ is the sigmoid function, by which we can transform the data to a value in the range of 0–1 to act as a gating signal, r_t controls the reset gate, and z_t controls the update gate. After obtaining the gated state, the reset gate r_t is first applied to the hidden layer output h^{t-1} at the previous moment to achieve the reset of the information state. Then, after multiplying it with the input text x^t together with the corresponding bias

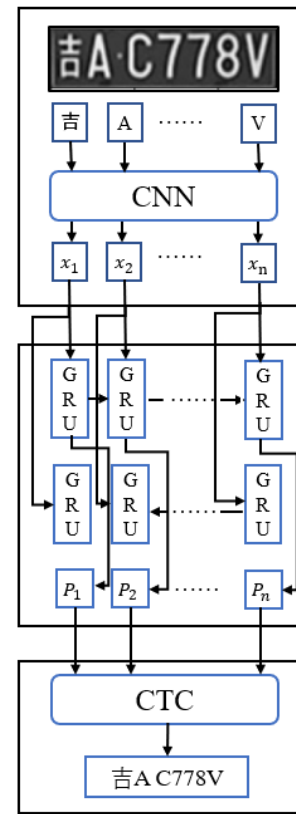


FIGURE 4. GRU + CTC recognition model.

and summing it, the update of the immediate information x^t of the node at the current moment is realized through the tanh activation.

$$h'_t = \tanh \left[W_{hh'} \left(r_t \cdot h^{t-1} \right) + W_{xh'} x^t + b_{h'} \right] \tag{11}$$

In the output of the hidden layer, GRU can use a single gated Z for both forgetting and selective memory. Yielding:

$$h^t = (1 - z) * h^{t-1} + z * h' \tag{12}$$

where $(1 - z)$ can be considered as the forgetting gate, which will clear the unimportant information in dimension h^{t-1} . $(1 - z) * h^{t-1}$ represents the selective clearing of the originally hidden state. $z * h'$ represents the selection of certain information in dimension h' . formula h^t is to clear certain dimensional information in h^{t-1} passed down and add certain dimensional information entered by the current node. The $*$ symbol here denotes Hadamard Product, which is a matrix multiplication method. The formula W_{xz} , W_{xr} , $W_{xh'}$ is the weight matrix of the connection between the hidden layers of the input layer at moment t , respectively; W_{hz} , W_{hr} , $W_{hh'}$ is the weight matrix of the connection between $t - 1$ and the hidden layers at moment t , respectively; b_z , b_r , $b_{h'}$ is the update gate, the reset gate, and the bias of the current hidden layer, respectively.

Finally, we use the Softmax layer to transform the state h^t of GRU into a probability distribution of 7 classes. In this paper, we use the challenging Chinese license plate as the

experimental object, and the Chinese license plate consists of seven characters, so we set seven characters information network layers.

$$p_t(c = c_k | x_t) = \text{softmax}(h_t), \quad k = 1, 2, \dots, 7 \quad (13)$$

The whole feature sequence will eventually be converted into probability estimation sequence ($p = p_1, p_2, \dots, p_L$), which has the same length as the output sequence.

4) SEQUENCE DECODING

In the final stage of the license plate recognition model, we convert the sequence of probability estimates P into a string. If the work is done using the common Softmax Loss, each column of output needs to correspond to a character element. This requires that each license plate image in the license plate training set needs to be labeled with the position of each character in the image, and then CNNs are used to find the alignment to each column of the Feature map to obtain the Label corresponding to that column output in order to train. However, in practice, it is very difficult to mark such aligned samples, and it is a huge job to mark the position of each character in addition to the marked characters. Moreover, the plate smudging and obscuring caused by complex environment may cause inconsistency in the number of plate characters, resulting in each column output not necessarily corresponding to each character one by one. Therefore, to solve the time series problem with uncertain alignment relationship between input features and output labels, we introduce CTC. It does not require data with pre-segmented and can directly decode the pre-reading sequence into output labels. Therefore, we connect CTC directly to the output of GRU, and the input of CTC happens to be the output activation of GRU. Additionally, we use the sequence decoding method to make better use of the output sequence of GRU and further obtain the optimal solution with maximum probability of approximate path.

IV. EXPERIMENTS AND RESULTS ANALYSIS

A. DATASET AND ENVIRONMENT

The license plate dataset used for the experiments in this paper is the open source large-scale Chinese urban parking dataset CCPD and some of the vehicle data images collected by ourselves, totaling 12500 images. The images in the dataset are all 1160×720 in size and contain images from many complex environments, such as bright light, cloudy sky, dark light, smudged license plates, tilted, etc., constituting a dataset with a diverse and sufficient number of license plate scenes distributed. We divide the dataset into training set and test set with the number of 10,000 and 2,500 respectively, which will be used to train the model and test the model effect. The data of the license plate recognition part requires us to detect and crop the license plate of the original image, and part of the sample data is shown in the Fig.5.

The experimental environment for this paper is built on a Linux operating system. The CPU model is an I7-12700H@4.7GHz, the GPU model is an NVIDIA GeForce



FIGURE 5. Sample images from the dataset.

RTX3090 8GB, and the software version is CUDA11.2 and PyTorch1.7.

B. EVALUATION INDICATORS

In this paper, we use objective evaluation criteria to evaluate license plate detection and recognition models. Precision, recall and mAP(mean Average Precision) are used as evaluation metrics with the following equations.

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

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

$$\text{mAP} = \frac{1}{C} \sum_{k=1}^n P(k) \Delta R(k) \quad (16)$$

where TP is the number of true positive samples, FP is the number of false positive samples, FN is the number of false negative samples, C is the number of categories, n is the number of referenced thresholds, k is the threshold, P(k) is the accuracy, and R(k) is the recall.

C. RESULTS ANALYSIS

1) LICENSE PLATE POSITIONING MODEL

In the training phase of the improved YOLOv5 license plate detection model, considering that we added the attention mechanism L-SE, which incorporates multi-level feature information, we set the batch size to 128, Epoch to 300, initial learning rate to 0.01, decay rate to 0.005, and iteration The total number of iterations is 30000, and the framework is Pytorch. Where batch size indicates the number of samples selected in a single round of training, and learning rate is the hyperparameter that knows how we should adjust the network weights by the gradient of the loss function. The purpose of license plate localization is to obtain the area where the license plate is located and provide data for license plate recognition. Therefore, the accuracy of license plate localization directly affects the effectiveness of license plate recognition, so we use Avg IOU, which responds to the accuracy of localization, to measure the effectiveness of license plate localization. The larger the value of this index, the better the positioning effect. The training result of the model is shown in Fig.6.

From (a) and (c) in Fig.6, it can be seen that after Epoch exceeds 50, the accuracy and mapping convergence of the model are between 0.90 – 0.99, which means that the model detection accuracy is high enough; In Figure b, when the recall convergence converges to 1 after 20 iterations,

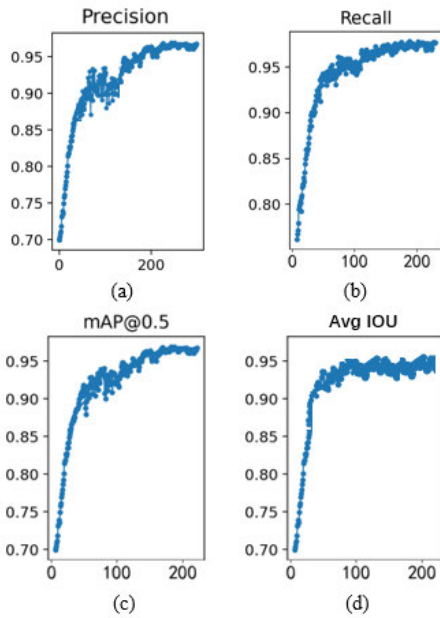


FIGURE 6. The training results of YOLOv5. (a) Precision (b) Recall (c) mAP (d) Avg IOU.

indicating that the target can be detected completely; figure d shows that as the training deepens, the AvgIOU of the model is stable between 0.9 and 1, indicating good model training results.

We compare the improved algorithm YOLOv5-LSE with YOLOv5-1 (only the class parameter is set to 1), YOLOv5, SSD300, Faster R-CNN, RPNNet and TE2E target detection algorithms for training tests. The comparison experiments will use the same training and test sets with the number of 10,000 for the former and 2,500 for the latter, and evaluate the performance of each mainstream algorithm using recall, precision, mAP and FPS. The results of the comparative tests are shown in Table 1. As we can see from the table, compared with the traditional target detection algorithms Faster R-CNN and SSD300, the YOLOv5 correlation algorithm is slightly deficient in detection speed, but the accuracy advantage is more obvious. Compared with the original YOLOv5 algorithm, the improved algorithm YOLOv5-LSE has increased complexity and slightly decreased in detection speed, but the recall, precision and mAP are improved by 3%, 4% and 2%, respectively, and the comprehensive performance is improved, and the detection speed FPS reaches the real-time requirement of engineering applications.

2) LICENSE PLATE RECOGNITION MODEL

The end-to-end license plate recognition model based on GRU+CTC uses 10,000 images, we set the batch_size is 128, the epoch is 30, the initial learning rate is 0.01 and the dynamic decay mechanism is used, and the gradient descent is optimized using the Adam algorithm and the framework is Pytorch. Compared with some traditional recognition algorithms, such as BP neural network [49], tesseract [50],

TABLE 1. Detection results of different algorithms.

Algorithm	Recall	Precision	mAP	Fps
Faster R-CNN [48]	86.5	89.4	92.3	15
SSD300 [49]	88.3	89.6	93.3	38
RPNnet [50]	95.2	94.8	94.2	62
YOLOv5	93.5	93.4	94.6	50
YOLOv5-1	95.4	95.2	95.8	46
YOLOv5-LSE	96.5	97.4	97.1	52

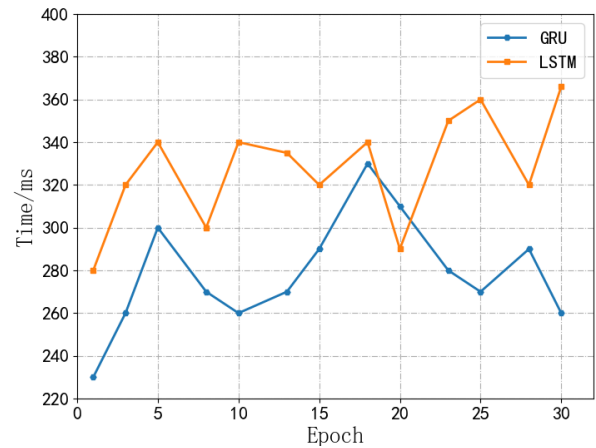


FIGURE 7. Model training time.

HOG + SVM [51], etc., the recognition effect advantage of deep learning-based algorithms is more obvious, so it is not compared with traditional algorithms in the recognition model validation stage. We choose to compare with the more popular and effective license plate recognition algorithm LSTM + CTC at this stage in the three directions of recognition precision, training time and CTC Loss to verify the cutting-edge and effectiveness of the recognition algorithm in this paper. At the end, we will use the complete end-to-end license plate recognition model including the target detection module to perform license plate detection and recognition in different practical scenarios to verify the precision and robustness of the license plate recognition system.

Fig.7 shows the training time of GRU + CTC compared to LSTM + CTC. From the figure, we can see that the training time of GRU + CTC is significantly lower than that of LSTM + CTC within the same number of Epochs, which is caused by the different network structures of the models themselves. The GRU network combines the input and forgetting gates in the LSTM network into one called the update gate. So that the GRU network itself contains only one surviving unit and updates reset two gates. GRU has one less gate function than LSTM, and the parameter size is 1/4 less than LSTM, and the computation is also greatly reduced, so the training time of GRU + CTC is less and the network converges faster. In the case of suitable amount of license plate data and all the hyper parameters are tuned, the performance of the two is comparable, and the structure of

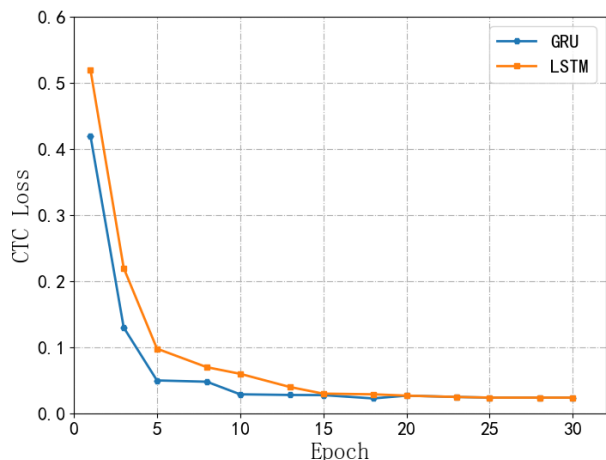


FIGURE 8. CTC Loss.

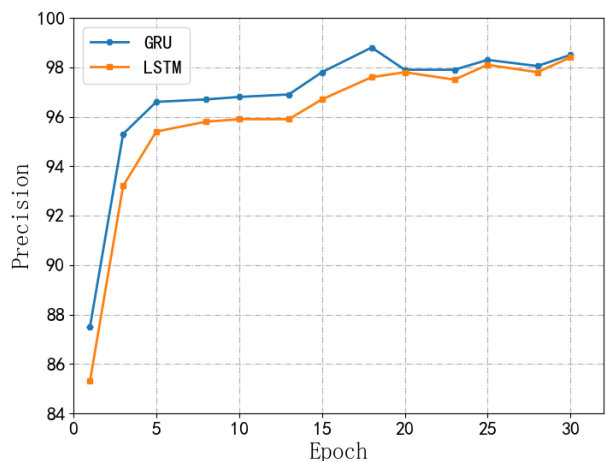


FIGURE 9. Model precision.

GRU is simpler, so GRU + CTC is chosen to be more efficient for training.

Fig.8 shows the CTC Loss variation curves for different network structures. From the figure, we can see that the CTC Loss values of GRU and LSTM converge from the 10th Epoch onward. However, in the initial training stage, the CTC Loss of GRU decreases faster. the CTC loss can be understood as the product of the probabilities of the corresponding labels of the output paths at each time step. Given an input, the output path probability here refers to the probability of the label of that input output at each time step from time $t = 1$ to T . The CTC loss essentially maximizes the probability sum of all paths, and the faster its value decreases before it stabilizes, the more accurate the model is represented.

Fig.9 shows the variation in recognition precision of the two recognition algorithms over the training cycle. From the figure, we can see that the model of GRU + CTC can obtain higher recognition precision in shorter time, and the precision reaches 96.6% at epoch = 8, while the precision of LSTM + CTC is only 95.4%, which is about 1.2% lower. In addition,

TABLE 2. Comparison of the effectiveness of complete LPR algorithms.

Model	framework	Precision(%)	Time(ms)
OPENCV + LSTM + CTC	PyTorch	96.94%	115.69
OPENCV + GRU + CTC	PyTorch	97.41%	102.32
YOLOv5 + LSTM + CTC	PyTorch	98.02%	69.47
YOLOv5 + GRU + CTC	PyTorch	98.32%	65.32
YOLOv5-LSE + LSTM + CTC	PyTorch	98.54%	73.25
YOLOv5-LSE + GRU + CTC	PyTorch	98.98%	70.25

the maximum accuracy of GRU + CTC is 98.8%, while the maximum accuracy of LSTM + CTC is 98.1%, which is slightly lower than the former by 0.7%. We calculated the average accuracy of the two recognition models, and the average accuracy of GRU + CTC is 97.6% and LSTM + CTC is 96.9%. Therefore, when the number of samples in the dataset is moderate, the GRU + CTC recognition model has high training efficiency and recognition effect.

Table 2 shows the comparison of recognition accuracy of the YOLOv5-LSE + GRU + CTC, the license plate recognition algorithm proposed in this paper, with other algorithms. In the comparison experiments, we not only chose the original YOLOv5 algorithm, but also added the OPENCV + RNN license plate recognition method. We use OPENCV to preprocess the image, a process that includes binarization, medianization, corruption, expansion and centralized blurring. After that, the rectangular feature aspect ratio (Ar) of the license plate is selected as the threshold to locate and segment the license plate, and the same algorithm as in this paper is chosen for the final character recognition module. From the table we can see that the license plate recognition model using deep learning localization algorithm has some advantages in terms of accuracy and recognition speed compared to the traditional localization algorithm model. Comparing the two localization algorithms, YOLOv5-LSE and OPENCV, we can see that with the same algorithm used in the recognition module, the experimental results show that the speed of license plate recognition using the YOLOv5-LSE localization algorithm is significantly higher and the recognition time is reduced by about 40%. In addition, when the localization algorithm of license plate is the same, the recognition module uses GRU + CTC, the recognition rate of license plate has some improvement, and because the structure of GRU is simpler compared with LSTM, it has obvious advantages in recognition speed. Then, using the improved YOLOv5-LSE for license plate location, because of the addition of the attention mechanism, which increases the complexity of the model, the recognition time increases by 4.93ms, but it has less impact on the overall performance of the license plate recognition model, and the improved algorithm improves 0.44% in recognition precision. In conclusion, the comprehensive performance of YOLOv5-LSE + GRU + CTC proposed in this paper is better and the model has the highest recognition precision.

The license plate recognition results using the YOLOv5-LSE license plate recognition model proposed in this paper



FIGURE 10. Recognition results with the proposed model.

under different practical scenarios and environmental conditions are shown in Fig. 10. Practical scenarios include streets, parking lots and highways, and environmental conditions include rain, strong light exposure and low light at night. Considering that the license plate recognition part is relatively small in the image, in order to show the recognition result as much as possible, we have partially cropped the result image and removed a little background part. As can be seen from the figure, our recognition model is able to accurately detect the license plate location and give the license plate character recognition results and the precision values. The sub-images (4, 5, 8) show the license plate recognition under different lighting scenes, and the experimental results show that the license plate recognition is accurate with recognition precision of 97.2%, 96.2% and 97.9%, respectively. Sub-images (1,3,6,7) show the license plate recognition under complex character conditions, which include consecutive identical characters, numbers and letters with similar shapes and Chinese abbreviations of different provinces in the license plate, and the experimental results show that the model has a good recognition effect. As shown above, the model proposed in this paper can accurately recognize license plate images in various scenes with strong stability and robustness.

V. CONCLUSION

Aiming at the problem that the traditional license plate recognition method lacking accuracy and speed, this paper proposes an end-to-end deep learning model for license plate localization and recognition in natural scenes. In our experiments, we worked on improving the deep learning-based YOLOv5 target detection algorithm by adding an improved attention mechanism LSE to its network structure. In addition, the number of parameters on the input side is reduced and only a single target class parameter is set in the YOLO layer. Finally, the recognition network is constructed using GRU + CTC to complete the recognition of license plates without character segmentation. The experimental results show that the improved YOLOv5-LSE + GRU + CTC license plate recognition model has obvious advantages over the traditional model, with an average recognition precision of 98.98% and FPS meeting the requirements of engineering applications. At the same time, through example verification, the improved model in this paper has good overall recognition effect in complex environment with strong stability and robustness. In our future work, we are committed to apply

the license plate recognition algorithm in the paper to the embedded system and realize the practical application work of the algorithm in life.

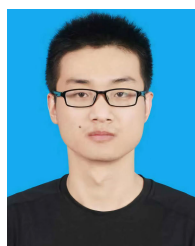
REFERENCES

- [1] A. Arshaghi, M. Ashourin, and L. Ghabeli, "Detection and classification of potato diseases potato using a new convolution neural network architecture," *Traitement Signal*, vol. 38, no. 6, pp. 1783–1791, Dec. 2021.
- [2] O. Bezsonov, O. Lebediev, V. Lebediev, Y. Megel, D. Prochukhan, and O. Rudenko, "Breed recognition and estimation of live weight of cattle based on methods of machine learning and computer vision," *Eastern-Eur. J. Enterprise Technol.*, vol. 6, no. 9, pp. 64–74, Dec. 2021.
- [3] K. Mohi-Alden, M. Omid, M. S. Firouz, and A. Nasiri, "Design and evaluation of an intelligent sorting system for bell pepper using deep convolutional neural networks," *J. Food Sci.*, vol. 87, no. 1, pp. 289–301, Jan. 2022.
- [4] J. Redmon, S. Divvala, and R. Girshick, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
- [5] M. Kim, J. Jeong, and S. Kim, "ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image," *Remote Sens.*, vol. 13, no. 23, p. 4851, Nov. 2021.
- [6] P. R. Miranda, D. Pestana, J. D. Lopes, R. P. Duarte, M. P. Véstias, H. C. Neto, and J. T. de Sousa, "Configurable hardware core for IoT object detection," *Future Internet*, vol. 13, no. 11, p. 280, 2021.
- [7] A. Mathias, S. Dhanalakshmi, and R. Kumar, "Occlusion aware underwater object tracking using hybrid adaptive deep SORT-YOLOv3 approach," *Multimedia Tools Appl.*, vol. 81, no. 30, pp. 44109–44121, Dec. 2022.
- [8] F. Lin, T. Hou, Q. Jin, and A. You, "Improved YOLO based detection algorithm for floating debris in waterway," *Entropy*, vol. 23, no. 9, p. 1111, Aug. 2021.
- [9] N. Zaghari, M. Fathy, S. M. Jameii, and M. Shahverdy, "The improvement in obstacle detection in autonomous vehicles using YOLO non-maximum suppression fuzzy algorithm," *J. Supercomput.*, vol. 77, no. 11, pp. 13421–13446, Nov. 2021.
- [10] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "YOLOv4-5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
- [11] K. Madasamy, V. Shanmuganathan, V. Kandasamy, M. Y. Lee, and M. Thangadurai, "OSDDY: Embedded system-based object surveillance detection system with small drone using deep YOLO," *EURASIP J. Image Video Process.*, vol. 2021, no. 1, pp. 1–14, Dec. 2021.
- [12] P. Tumas, A. Serackis, and A. Nowosielski, "Augmentation of severe weather impact to far-infrared sensor images to improve pedestrian detection system," *Electronics*, vol. 10, no. 8, p. 934, Apr. 2021.
- [13] G. Li, Z. Ji, X. Qu, R. Zhou, and D. Cao, "Cross-domain object detection for autonomous driving: A stepwise domain adaptive YOLO approach," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 603–615, Sep. 2022.
- [14] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*.
- [15] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, "CNN-RNN based method for license plate recognition," *CAAI Trans. Intell. Technol.*, vol. 3, no. 3, pp. 169–175, 2018.
- [16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. O. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, 2015, 2048–2057.
- [17] L. Feng, C. Cheng, M. Zhao, H. Deng, and Y. Zhang, "EEG-based emotion recognition using spatial-temporal graph convolutional LSTM with attention mechanism," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 11, pp. 5406–5417, Nov. 2022.
- [18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.
- [19] H. Wang, Y. U. Zhu, B. Green, H. Adam, A. Y. Uille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," 2020, *arXiv:2003.07853*.
- [20] Z. Tang Yushun, "Color-based license plate recognition research," *Mod. Comput.*, vol. 32, no. 71, pp. 63–66, 2020.
- [21] X. Li, "Extraction of Macau car license plate location based on image texture recognition," *Inf. Technol. Informatization*, vol. 11, pp. 237–240, 2020, doi: 10.3969/j.issn.1672-9528.2020.11.074.

- [22] L. Xiang and H. Bang, "Multi—License plate location based on improved two—Dimensional discrete wavelet transform," *J. Chongqing Jiaotong Univ., Natural Sci.*, vol. 39, no. 2, pp. 16–21, 2020.
- [23] Z. Zhenwei and S. Yanchen, "Location method for license plate based on M—HSI and pixel offset," *Comput. Eng. Des.*, vol. 39, no. 11, pp. 3576–3583, 2018, doi: 10.16208/j.issn1000-7024.2018.11.049.
- [24] P. F. Felzenszwalb, R. B. Girshick, and D. Mcallester, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [25] M. Ravichandran and S. Sumitha, "Robust automated license plate and character recognition," *Digit. Image Process.*, vol. 4, no. 4, 2012. [Online]. Available: <http://www.citresearch.org/dl/index.php/dip/article/view/DIP032012009>
- [26] Y. Tian, J. Song, X. Zhang, P. Shen, L. Zhang, W. Gong, W. Wei, and G. Zhu, "An algorithm combined with color differential models for license-plate location," *Neurocomputing*, vol. 212, pp. 22–35, Nov. 2016.
- [27] A. O. Salau, T. K. Yesufu, and B. S. Ogunbare, "Vehicle plate number localization using a modified GrabCut algorithm," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 33, no. 4, pp. 399–407, May 2021.
- [28] Z. Mahmood, K. Khan, U. Khan, S. H. Adil, S. S. A. Ali, and M. Shahzad, "Towards automatic license plate detection," *Sensors*, vol. 22, no. 3, p. 1245, Feb. 2022.
- [29] S. Sweta, "Automatic car license plate detection system," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1116, no. 1, 2021, doi: 10.1088/1757-899X/1116/1/012198.
- [30] N. Omar, A. Sengur, and S. G. S. Al-Ali, "Cascaded deep learning-based efficient approach for license plate detection and recognition," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113280.
- [31] I. Slimani, A. Zaarane, W. Al Okaishi, I. Atouf, and A. Hamdoun, "An automated license plate detection and recognition system based on wavelet decomposition and CNN," *Array*, vol. 8, Dec. 2020, Art. no. 100040.
- [32] A. Mensch, G. Varoquaux, and B. Thirion, "Compressed online dictionary learning for fast resting-state fMRI decomposition," in *Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2016, pp. 1282–1285.
- [33] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [34] Y. Zou, Y. Zhang, J. Yan, X. Jiang, T. Huang, H. Fan, and Z. Cui, "License plate detection and recognition based on YOLOv3 and ILPRNET," *Signal, Image Video Process.*, vol. 16, no. 2, pp. 473–480, Mar. 2022.
- [35] Y. Sun, Q. Peng, and D. Zhang, "Light-YOLOv3: License plate detection in multi-vehicle scenario," *IEICE Trans. Inf. Syst.*, vol. E104.D, no. 5, pp. 723–728, May 2021.
- [36] S. S. Tabrizi and N. Cavus, "A hybrid KNN-SVM model for Iranian license plate recognition," *Proc. Comput. Sci.*, vol. 102, pp. 588–594, Jan. 2016.
- [37] B. Pu, "Research on Chinese license plate recognition algorithm based on convolution neural network," *J. Phys., Conf. Ser.*, vol. 1802, no. 3, Mar. 2021, Art. no. 032055.
- [38] N. M. Shatnawi, "Bees algorithm and support vector machine for Malaysian license plate recognition," *Int. J. Bus. Inf. Syst.*, vol. 28, no. 3, p. 284, 2018.
- [39] X. Kong, K. Wang, M. Hou, X. Hao, G. Shen, X. Chen, and F. Xia, "A federated learning-based license plate recognition scheme for 5G-enabled Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8523–8530, Dec. 2021.
- [40] S. Alghyaline, "Real-time Jordanian license plate recognition using deep learning," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2601–2609, Jun. 2022.
- [41] P. Liu, G. Li, and D. Tu, "Low-quality license plate character recognition based on CNN," in *Proc. 8th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2015, pp. 53–58.
- [42] M. C. Rademeyer, A. Barnard, and M. J. Booysen, "Optoelectronic and environmental factors affecting the accuracy of crowd-sourced vehicle-mounted license plate recognition," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 15–28, 2020.
- [43] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation," 2020, *arXiv:2003.07853*.
- [44] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
- [45] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13713–13722.
- [46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [47] W. Liu, D. Anguelov, and D. Erhan, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. Heidelberg, Germany: Springer*, 2016, pp. 21–37.
- [48] Z. B. Xu, W. Yang, and A. J. Meng, "Towards end-to-end license plate detection and recognition: A large dataset and baseline," in *Proc. Eur. Conf. Comput. Vis. Heidelberg, Germany: Springer*, 2018, pp. 261–277.
- [49] X. C. Wang, G. W. Yang, and Y. Yang, "License plate fault-tolerant characters recognition algorithm based on color segmentation and BP neural network," *Appl. Mech. Mater.*, vols. 411–414, pp. 1281–1286, Sep. 2013.
- [50] R. Polshetty, M. Roopaei, and P. Rad, "A next-generation secure cloud-based deep learning license plate recognition for smart cities," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 286–293.
- [51] R. Islam, M. R. Islam, and K. H. Talukder, "An efficient method for extraction and recognition of Bangla characters from vehicle license plates," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 20107–20132, Jul. 2020.



HENGLIANG SHI received the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, China. He was an Associate Professor and the Dean of the School of Automotive and Rail Transportation, Luoyang Polytechnic. His research interests include video tracking, intelligent detection, and big data analysis.



DONGNAN ZHAO is currently pursuing the degree with the Henan University of Science and Technology, China. His research interests include deep learning, computer vision, and big data analysis.