Research on license plate recognition based on graphically supervised signal-assisted training (#108697)

First submission

Guidance from your Editor

Please submit by **1 Apr 2025** for the benefit of the authors (and your token reward) .

Structure and Criteria
Please read the 'Structure and Criteria' page for guidance.

Raw data check
Review the raw data.

Image check
Check that figures and images have not been inappropriately manipulated.

If this article is published your review will be made public. You can choose whether to sign your review. If uploading a PDF please remove any identifiable information (if you want to remain anonymous).

Files
Download and review all files from the materials page.

14 Figure file(s)
5 Table file(s)

Structure and Criteria

Structure your review

The review form is divided into 5 sections. Please consider these when composing your review:

1. **BASIC REPORTING**
2. **EXPERIMENTAL DESIGN**
3. **VALIDITY OF THE FINDINGS**
4. General comments
5. Confidential notes to the editor

You can also annotate this PDF and upload it as part of your review

When ready submit online.

Editorial Criteria

Use these criteria points to structure your review. The full detailed editorial criteria is on your guidance page.

BASIC REPORTING

- Clear, unambiguous, professional English language used throughout.
- Intro & background to show context. Literature well referenced & relevant.
- Structure conforms to PeerJ standards, discipline norm, or improved for clarity.
- Figures are relevant, high quality, well labelled & described.
- Raw data supplied (see PeerJ policy).

EXPERIMENTAL DESIGN

- Original primary research within Scope of the journal.
- Research question well defined, relevant & meaningful. It is stated how the research fills an identified knowledge gap.
- Rigorous investigation performed to a high technical & ethical standard.
- Methods described with sufficient detail & information to replicate.

VALIDITY OF THE FINDINGS

- **Impact and novelty is not assessed.** Meaningful replication encouraged where rationale & benefit to literature is clearly stated.
- All underlying data have been provided; they are robust, statistically sound, & controlled.
- Conclusions are well stated, linked to original research question & limited to supporting results.

Standout reviewing tips

The best reviewers use these techniques

Tip	*Example*
Support criticisms with evidence from the text or from other sources	*Smith et al (J of Methodology, 2005, V3, pp 123) have shown that the analysis you use in Lines 241-250 is not the most appropriate for this situation. Please explain why you used this method.*
Give specific suggestions on how to improve the manuscript	*Your introduction needs more detail. I suggest that you improve the description at lines 57- 86 to provide more justification for your study (specifically, you should expand upon the knowledge gap being filled).*
Comment on language and grammar issues	*The English language should be improved to ensure that an international audience can clearly understand your text. Some examples where the language could be improved include lines 23, 77, 121, 128 – the current phrasing makes comprehension difficult. I suggest you have a colleague who is proficient in English and familiar with the subject matter review your manuscript, or contact a professional editing service.*
Organize by importance of the issues, and number your points	*1. Your most important issue* *2. The next most important item* *3. …* *4. The least important points*
Please provide constructive criticism, and avoid personal opinions	*I thank you for providing the raw data, however your supplemental files need more descriptive metadata identifiers to be useful to future readers. Although your results are compelling, the data analysis should be improved in the following ways: AA, BB, CC*
Comment on strengths (as well as weaknesses) of the manuscript	*I commend the authors for their extensive data set, compiled over many years of detailed fieldwork. In addition, the manuscript is clearly written in professional, unambiguous language. If there is a weakness, it is in the statistical analysis (as I have noted above) which should be improved upon before Acceptance.*

Research on license plate recognition based on graphically supervised signal-assisted training

Dianwei Chi [Corresp., 1], **Zehao Jia** [Corresp., 1], **Lizhen Liu** [2]

[1] Artificial Intelligence Institute, Yantai Institute of Technology, Yantai, Shandong, China

[2] Institute of Microbiology, Jiangxi Academy of Sciences, Nanchang, Jiangxi, China

Corresponding Authors: Dianwei Chi, Zehao Jia
Email address: dianwei.chi@163.com, 19863817078@163.com

Background. With the rapid growth of the number of cars and the increasing complexity of urban transportation, it is particularly important to achieve high-accuracy license plate recognition in complex scenarios. However, since license plate recognition models are mostly deployed on embedded devices with limited computational resources, designing a lightweight and accurate model has become an urgent problem in the field of license plate recognition.

Methods. In this study, we propose a license plate recognition algorithm that uses LPRNet (License Plate Recognition Network) as the base model and incorporates graphically supervised signals for assisted training, aiming to improve the accuracy of the model by improving its training process, and to realize a lightweight and highly accurate license plate recognition model. The study adds an auxiliary training branch to the conventional training, using graphically supervised signals to guide the model to learn image features.

Results. Experiments show that compared with LPRNet, this study improves the accuracy in all test sets of the CCPD dataset, where the average accuracy is improved by 5.86%, the maximum accuracy by 10.9%, the average character precision by 2.1%, and the average recall by 6.9%, indicating that this study can achieve higher accuracy while keeping it lightweight. This study also provides new ideas for other deep learning image recognition tasks.

Research on license plate recognition based on graphically supervised signal-assisted training

Dian-Wei Chi[1,*], Ze-Hao Jia[1,*], Li-Zhen Liu[2]

[1] School of Artificial Intelligence, Yantai Institute of Technology, Yantai 264003, China
[2] Institute of Microbiology, Jiangxi Academy of Sciences, Nanchang 330096, China

Corresponding Author:
Dian-Wei Chi[1] and Ze-Hao Jia[1]
Street Address, City, State/Province, Zip code, Country
Email address: dianwei.chi@163.com and 19863817078@163.com

Abstract

Background. With the rapid growth of the number of cars and the increasing complexity of urban transportation, it is particularly important to achieve high-accuracy license plate recognition in complex scenarios. However, since license plate recognition models are mostly deployed on embedded devices with limited computational resources, designing a lightweight and accurate model has become an urgent problem in the field of license plate recognition.
Methods. In this study, we propose a license plate recognition algorithm that uses LPRNet (License Plate Recognition Network) as the base model and incorporates graphically supervised signals for assisted training, aiming to improve the accuracy of the model by improving its training process, and to realize a lightweight and highly accurate license plate recognition model. The study adds an auxiliary training branch to the conventional training, using graphically supervised signals to guide the model to learn image features.
Results. Experiments show that compared with LPRNet, this study improves the accuracy in all test sets of the CCPD dataset, where the average accuracy is improved by 5.86%, the maximum accuracy by 10.9%, the average character precision by 2.1%, and the average recall by 6.9%, indicating that this study can achieve higher accuracy while keeping it lightweight. This study also provides new ideas for other deep learning image recognition tasks.
Keywords: deep learning; supervised signaling; assisted training; license plate recognition

Introduction

With the rapid growth of the number of cars and the increasing complexity of urban transportation, vehicle management and traffic monitoring are particularly important. Thanks to the development of the Internet and deep learning technology, smart transportation has become a major trend in vehicle management and traffic monitoring (Wu, Wu and Wang, 2019; Yang, Xie and Hu, 2024). As a key link in the intelligent transportation system, license plate recognition

technology is of great significance for the realization of vehicle violation monitoring, parking lot management, traffic congestion analysis, etc.

Deep learning methods are highly regarded for their powerful feature learning abilities and efficient modeling of complex data. With deep learning models, the system can automatically learn the character features on the license plate from a large amount of data and can accurately recognize them under different lighting, angles, and even partial occlusion. Compared with traditional methods, deep learning methods show higher robustness and accuracy in the field of license plate recognition and can effectively cope with the challenges in complex environments. At the same time, most of the application scenarios for license plate recognition are embedded devices with limited computational resources, which requires the deep learning model to be lightweight while improving performance.

For research on deep learning license plate recognition, the traditional approach to license plate recognition is usually performed in a two-stage process: first character segmentation of the license plate, followed by recognition of individual characters. For example, Zhang et al. (2018) used a morphological algorithm to locate the license plate region and segmented the license plate characters using a vertical projection-based method. Khan et al. (2018) combined the features of the image to segment the characters and used a support vector machine to classify the characters. Raza M et al. (2020) used color space with a vertical projection method for license plate recognition and an improved deep learning model for single character recognition, which improved the accuracy, but the recognition results were still limited by the character segmentation results. This way of doing character segmentation first and then single-character recognition not only has low accuracy but also low detection efficiency.

Subsequently, a series of tasks for license plate recognition via neural networks emerged. Shi et al. (2016) proposed the CRNN model, which solves the problem of end-to-end license plate recognition by considering the problem such as license plate recognition as a multi-input and multi-output image sequence recognition problem. Zherzdev and Gruzdev (2018) proposed the LPRNet model, which used purely convolutional operations for the first time to recognize license plates.

In recent years, in order to improve the adaptability of the recognition model for complex scenes, some researchers have proposed a license plate recognition model that incorporates denoising structures. For example, the license plate recognition system designed by Rao Wenjun et al. (2021) achieved better license plate detection and recognition accuracy by improving the CNN detection network, STN, and bidirectional recurrent neural network (BRNN). Pirgazi J et al. (2018) combined convolutional neural network with LSTM to achieve end-to-end license plate recognition but with low computational efficiency. Caizhen Zhang et al. (2021) changed the standard CNN in CRNN to a micro-modified model of a depth-separable convolutional network and used a bidirectional long- and short-term memory network for RNN to improve the recognition accuracy of the model. Xiangpeng Li et al. (2019) proposed an enhanced convolutional neural network model, AlexNet-L, which is an end-to-end network model for license plate character recognition, to improve the accuracy of license plate recognition. Wang

and Chen (2024) proposed a special license plate recognition algorithm that uses deep residual networks and attention mechanisms to optimize the model structure and improve recognition accuracy. Kim et al. (2024) proposed a new license plate recognition method, AFA-Net, which achieves clarity restoration and significantly improves the recognition accuracy of feature attention networks for low-resolution and motion blurred images. Xu F. et al. (2024) proposed a CRNN-based method for ship license plate image recognition, improved the CRNN model through image enhancement and data enhancement, and achieved an accuracy of 93% on a wide range of image datasets. Lee et al. (2019) proposed an algorithm called SNIDER for license plate recognition from low-quality images, which integrates denoising and rectification tasks through a training network framework, effectively improving the model's performance. Zhang, Liu and Ma (2018) combined license plate super-resolution and recognition in an end-to-end framework to improve the accuracy and adaptability of the model. Min et al. (2020) proposed the Fast-SRGAN model, which uses super-resolution to remove the noise present in the image and improve the final performance. Xu H. et al. (2021) proposed a 3D perspective transform-based License Plate Correction Algorithm APAN, which improves recognition accuracy through a GRU structure based on a 2D attention mechanism. Ceng G. and Ke X. (2021) introduced Vision Transformer into the study of license plate recognition, which suppresses noise and improves accuracy through self-attention mechanisms. In addition, some studies improve the model's adaptability in complex situations through data augmentation. For example, Yang, Wang and Jiang (2024) utilized a haze enhancement algorithm to improve the recognition accuracy of the model in haze scenarios. Zhang et al. (2020) proposed a CycleGAN model for generating license plate images and a license plate recognizer based on an Xception-based CNN encoder with a 2D attention mechanism, which experimentally proved to have good performance in a variety of environments. Jin et al. (2021) applied a dark channel a priori algorithm to preliminary dehaze fuzzy images and realized image super-resolution by a super-resolution convolutional neural network to improve the accuracy in hazy scenes. Zhu S et al. (2024) improved the YOLOv5 algorithm to improve the accuracy of license plate localization but did not recognize license plate characters.

In summary, current deep learning models have achieved significant performance improvements on license plate recognition tasks, but these improvements often come at the cost of increasing the number of parameters and computational complexity of the model. This approach not only increases the storage and running cost of the model, but also becomes impractical to deploy on resource-constrained embedded devices. Although some studies have proposed lighter model structure designs, such as LPRNet and CRNN, which are able to be deployed in embedded devices, the insufficient learning ability and generalization of the overly lightweight models result in lower accuracy rates.

Therefore, how to further improve the accuracy of models while keeping them lightweight has become an urgent problem in the field of license plate recognition. This study focuses on optimizing the training process of existing lightweight models to improve their recognition accuracy and generalization ability.

In this paper, we propose a research on license plate recognition based on graphical supervised signal assisted training, by introducing the labeling of image modality and auxiliary training branch to assist the training of the model, the auxiliary training branch only plays a role in the training phase, and the relevant structure of the auxiliary training branch will be removed during the testing phase, in order to ensure that the model is lightweight. This research effectively improves the accuracy of the lightweight model.

The contributions of this paper are as follows:

1. The supervisory signals carried in text information are explored and investigated to mine a new graphical supervisory signal.
2. The graphical supervisory signal carried in the text is applied to the auxiliary training of the license plate recognition neural network, which is effective.
3. Discusses the universality and advantages of graphical supervised signal-assisted training, providing more possibilities for it in the field of deep learning.

The remainder of the article is organized as follows. Section 2 briefly introduces the related work. Section 3 describes the design flow of the methodology of this paper. Section 4 evaluates the methodology of this paper through experiments. Section 5 concludes the paper and outlines future research directions.

Related work

CTC Loss

In the field of license plate recognition, CTC (Connectionist Temporal Classification) Loss is widely used to solve the no-alignment supervised problem and is the key to achieve the end-to-end license plate recognition problem. Character sequences in license plate recognition tasks often have variable lengths and complex alignments, especially in the absence of character level annotations, which makes it difficult to apply traditional sequence classification methods. CTC aims to solve the character recognition and alignment problem by modeling the inconsistency in the lengths of the input and output sequences.

Suppose the input sequence of CTC is $X = (x_1, x_2, ..., x_T)$, where T is the CTC input length. the input of CTC is the output of the license plate recognition model, which is a fixed length. Suppose the output sequence of CTC is $Y = (y_1, y_2, ..., y_L)$, where L is the CTC output length and $L \leqslant T$. The purpose of CTC is to find a mapping from X to Y. CTC generates alignment paths by 1. inserting a blank character ∅, at each position in the sequence of output characters to indicate that no character is output at a certain time step. 2. allowing the neighboring output characters to remain the same to indicate continuous output of characters. With these operations, CTC can align fixed-length input sequences with shorter and variable-length output sequences. The output of CTC is actually a probability distribution of a series of labeled sequences, each of which can generate different alignment paths by inserting blank characters and repeating characters. The final output is the probability sum of all possible paths.

The goal of CTC Loss is to maximize the total probability of an alignment path for a sequence Y of correct characters. Let B denote the function that maps alignment paths to output sequences

and $P(Z|X)$ denote the probability of generating alignment path Z given input X, The CTC Loss can be expressed as:

$$P(Y|X) = \sum_{Z \in B^{-1}(Y)} P(Z|X) \quad (1)$$

Where, $B^{-1}(Y)$ denotes the set of all alignment paths that can be mapped to the output sequence Y. To compute this probability, CTC computes the probability of each aligned path and sums it by means of dynamic programming.

After calculating the probabilities of all possible paths, the final form of the CTC Loss is the negative log-likelihood function:

$$\text{CTC Loss} = -\log P(Y|X) \quad (2)$$

The advantage of CTC Loss is that sequence alignment can be performed directly in the training phase without additional alignment labels. It can effectively deal with sequence alignment problems and can automatically learn the alignment corresponding to the target sequence during the training process. In license plate recognition, CTC Loss can be used to train neural network models to improve the accuracy of license plate recognition by optimizing CTC Loss. By introducing CTC Loss, the license plate recognition model can learn the position of the characters and the alignment relationship between the characters, thus improving recognition accuracy.

CTC Loss is the key to achieve end-to-end license plate recognition and is the main loss function in the license plate recognition process. In this paper, we optimize the output of CTC Loss by introducing auxiliary training to assist CTC Loss for training.

LPRNet

Conventional sequence recognition problems are usually implemented using RNN neural networks, but the operation of license plate recognition using RNNs is not very efficient due to the specificity of the RNN structure. LPRNet is one of the widely used models in the field of license plate recognition, and its performance on the test set is excellent, while the model is lightweight enough for embedded device deployment. Therefore, we choose LPRNet as the base framework to ensure the balance between high recognition accuracy and low computational resource consumption.

LPRNet is a deep learning network model specially designed for license plate recognition, which is a very efficient neural network that requires only 0.34 GFLOPs for a single forward propagation. LPRNet does not require pre-segmentation of characters, and is characterized by high accuracy, real-time performance, and support for variable-length character license plate recognition. For license plates from different countries with large character differences, LPRNet is capable of end-to-end training. Moreover, as the first real-time lightweight license plate recognition algorithm that does not use RNN, LPRNet can run on various devices, including embedded devices.

The backbone network of LPRNet takes the original RGB image as input, extracts image features using a convolutional neural network (CNN), and replaces the RNN neural network with a context-bound 1*13 convolutional kernel, the structure of which is shown in Fig. 1. In addition,

global context embedding is added to the intermediate feature mapping of the classifier to further improve the expressive power of the neural network. Similar to RNN, LPRNet uses CTC Loss for end-to-end training because the output coding of the network is not equal to the length of the license plate characters.

In the field of license plate recognition, the LPRNet model stands out with its relatively high accuracy and lightweight design, and has become one of the most widely used models in the field. Its efficient computational performance and low resource consumption have led to its successful deployment in many embedded devices. In view of this, LPRNet is chosen as the base model in this study, aiming to further improve its recognition accuracy through in-depth optimization of its training process.

Methods

Graphically supervised signals

In the context of big data, more and more research is looking at how to effectively utilize supervised signals. Many studies on Self-supervised Learning have emerged in recent years. In the field of machine vision, many studies focus on mining supervised signals from the data itself, among which the more typical one is self-supervised learning. For example, He et al. (2022) proposed the MoCo algorithm, which performs unsupervised learning by constructing a dynamic dictionary in the form of contrast learning. Chen et al. (2020) proposed the SimCLR algorithm, which improves the effect of contrast learning by data augmentation with the construction of larger batches. Chen and He (2021) further elaborated the self-supervised algorithm, which solves the problem of gradient collapse by using gradient stopping. He et al. (2022) again re-explored the supervised signals in the image by first randomly masking the image and then predicting the masked portion from the unmasked portion. In the field of natural language processing (Devlin et al., 2018; Radford et al., 2018), although not using the idea of self-supervised learning, it is similar to self-supervised learning, i.e., the model is first trained by a large-scale pre-training task, which allows the model to acquire a certain amount of learning ability in advance by pre-training on an unsupervised, large-scale textual dataset, before fine-tuning it on a specific task.

The license plate recognition task has some special characteristics. Firstly, the resolution of license plate images is small, and small resolution license plate images cannot adapt to most of the auxiliary tasks, which makes self-supervised learning difficult to realize in license plate images; secondly, the signal-to-noise ratio of license plate images in complex scenes is lower compared to that of conventional images, and conventional auxiliary tasks may make the model learn more noise, which in turn reduces the performance of the model.

The research in this paper is inspired by self-supervised learning, hoping to mine some additional valuable information in the existing dataset and use this additional information to assist the training of the model, in order to improve the model's performance at the time of testing. However, the research in this paper is not exactly the same as self-supervised learning, and the

difference between the research in this paper and self-supervised learning is specifically manifested in the following two aspects:
1. the supervisory information of self-supervised learning comes from data, while the supervisory information of this study comes from labels.
2. self-supervised learning mostly uses the training paradigm of unsupervised pre-training-supervised fine-tuning, while this study merges the pre-training process with fine-tuning, and performs pre-training and fine-tuning at the same time in the form of dual training branches to ensure the effectiveness of the model.

In this paper, we hope to mine new supervisory signals using known information for better results in the license plate recognition task. In traditional deep learning-based license plate recognition research, supervised learning is usually performed. Taking LPRNet as an example, it takes RGB images as the input of the neural network, fixed-length feature sequences as the output of the neural network, and variable-length text sequences as the labels for neural network training. Through the coordination of CTC Loss, the output of the fixed-length feature sequence and the label of the variable-length text sequence are matched, and back propagation is completed to update the parameters of the neural network. At this time, the supervisory signal of the neural network comes from the text sequence labels, and the updating of the parameters of the neural network depends entirely on the backpropagation of the CTC Loss.

The conventional use of text sequences for labeling can be regarded as a semantic-level supervisory signal, which guides the model to understand the semantics in the image as much as possible. In this paper, we hope to find a method that can assist the model in learning the semantic features better and thus improve the correctness as well as the robustness of the model. This paper mines the graphical supervisory signals carried in the text label to enhance the learning effect of the neural network. That is, the graphical features carried in the text information are used as another supervisory signal to assist the neural network to better learn the semantic features of the image and achieve better classification results.

For license plate recognition or similar text recognition problems, we find that the text label contains graphical features. Specifically, the graphical features in text label come from "fonts". Fonts can transform text signals into specific graphical features, so the text information can be constructed into an image through fonts, and through the constructed image, the neural network can achieve better learning results.

In the conventional license plate recognition problem, string labels are often mapped to indexes through word lists, participating in the training of neural networks in the form of ONE HOT encoding, at which time the supervised signals are supervised signals from the semantic level. And through the font, the string can be constructed as an image; the graphical features of the character are completely reflected in the image label, so the supervised signal at this time can be regarded as the supervised signal from the graphical level, as shown in Fig. 2.

The image label of license plate is constructed precisely through fonts, and graphical supervisory signals are introduced in an attempt to obtain supervisory signals in fonts with respect to the graphical features of the text. As shown in Fig. 3, in constructing the image label, we spatially

aligns the real training image with the image label. The goal is to make the location of the license plate pixels in the image label as close as possible to the location of the license plate pixels in the original map.

The input to the spatial alignment algorithm is the average image of the training set, and the output is the alignment boundary. The training set is first traversed to find the average image of all license plates, using the average map as an abstract representation of the entire training set. Subsequently, the average map is processed to remove irrelevant information from the image by grayscaling and binarization. Next, the image is projected, respectively, on the horizontal axis and the vertical axis of the two directions. The two projection curves, respectively, set the threshold values of the two directions, and two threshold straight lines can be obtained. Through the threshold straight line and the intersection of the projection curve, you can determine the spatial alignment of the boundary. Then determine the alignment boundary. You can generate the same size with the boundary of the characters, which will be placed in the image in accordance with the alignment boundary.

Auxiliary training

Upsampling decoder

How to effectively utilize graphically supervised signals is a central issue in the task flow. This paper draws on the ideas in autoencoder (Michelucci,, 2022). Autoencoder belong to unsupervised learning, which predicts the input data through the output data so that the output is as similar as possible to the input, and the purpose of autoencoder is generally to compress the input data by using the hidden layer. Autoencoder put the supervised signals they carry to good use by predicting the input. For the problem of how to utilize the graphical supervisory signal, this paper refers to the idea of a autoencoder. The difference is that the autoencoder predicts the input, while this paper predicts the image label.

In order to effectively utilize the image label as a supervised signal for training, we design an upsampling decoder in the feature output layer of the original neural network. The construction of the upsampling decoder is accomplished using only a simple three-layer transposed convolution and a projected convolution. The input to the upsampling decoder is the feature map output from the LPRNet backbone, which is used as the input to both the text sequence classifier and the upsampling decoder. The location of the upsampling decoder and its structure are shown in Fig. 4. After processing by the upsampling decoder, the feature map is sampled to the same size as the image label. Subsequently, the output of the upsampling decoder is made as close as possible to the image label by MSE Loss.

MSE Loss in upsampling decoder can be expressed as:

$$L_{mse} = \frac{1}{n \times h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} (y_{i,j} - \hat{y}_{ij})^2 \quad (3)$$

Where, n denotes the number of license plate samples, h and w respectively denotes the height and width of the image label, y denote the outputs of upsampling decoder, y denote image label. The closer the upsampling decoder output is to the image label, the smaller the loss.

Algorithmic process

The current conventional license plate recognition process is to input the RGB image of the license plate, get the output of the category sequence through the deep learning model, and then match the category sequence with the label sequence through CTC Loss and calculate the loss. The difference in the algorithm flow in this paper is that a new auxiliary training branch is designed, and the auxiliary training branch is used to receive graphic supervised signals. The training process of the model consists of the auxiliary training branch and the regular training branch, and the training process is shown in Fig. 5. The models in the regular training branch and the auxiliary training branch are slightly different, and this paper calls the model in the regular training branch the regular model and the model in the auxiliary training branch the auxiliary model. The inputs of the regular model and the auxiliary model are exactly the same, and the backbone part of the model is weight-sharing, which means that the same backbone completes the two training tasks of the two training branches at the same time, i.e., the two training branches are training the same backbone. The difference between the regular model and the auxiliary model is that the regular model inputs the output of the backbone directly into the classifier to get the category sequence, while the auxiliary model inputs the input out of the backbone into the upsampling decoder, which decodes the feature map output from the backbone into an image of the same size as the image label to receive the graphical supervisory signals. During model training, the two training branches alternate. That is, for a certain batch of data in the training set, the model weights are first updated through the regular training branch, and then using the same data, the model weights are updated through the auxiliary training branch. When both branches have finished updating the weights, the next batch of data is obtained, and the above process is repeated. In the testing phase, only the regular training branch is retained, and the auxiliary training branch is no longer needed. Therefore, no additional computation is added to the model in the testing phase.

In order to prove that the algorithmic process effectively utilizes the graphically supervised signals, using θ to denote the neural network weights, x to denote the image input, y_{ctc} to denote the sequence labels, y_{mse} to denote the image labels (graphically supervised signals), and $f_\theta(x)$ to denote the neural network model, the loss function can be expressed as:

$$L_{mse}(\theta) = \text{MSE Loss}(f_\theta(x), y_{mse}) \tag{4}$$

$$L_{ctc}(\theta) = \text{CTC Loss}(f_\theta(x), y_{ctc}) \tag{5}$$

Using α to denote the learning rate, the weight update formula can be expressed as:

$$\theta_{mse\ t} = \theta_{mse\ t-1} - \alpha \nabla L_{mse}(\theta) \tag{6}$$

$$\theta_{ctc\ t} = \theta_{ctc\ t-1} - \alpha \nabla L_{ctc}(\theta) \tag{7}$$

In the auxiliary training process, the regular training branch and the auxiliary training branch are carried out alternately, and the weights of the model are shared, so $\theta_{ctc\ t-1} = \theta_{mse\ t}$, then:

$$\theta_{ctc\ t} = (\theta_{mse\ t-1} - \alpha \nabla L_{mse}(\theta)) - \alpha \nabla L_{ctc}(\theta) \quad (8)$$

$$\theta_{ctc\ t} = \theta_{mse\ t-1} - (\alpha \nabla L_{mse}(\theta) + \alpha \nabla L_{ctc}(\theta)) \quad (9)$$

Since the prediction phase relies on the regular training branch of the model, $\theta_{ctc\ t}$ is used to represent the final result. According to formula (9), the parameters are updated not only on the gradient returned by CTC Loss but also on the gradient returned by MSE Loss. Therefore, it can be proved that the graphical supervision signal is effectively applied to the weight update of the conventional model.

Experiments and Discussions

Training dataset and evaluation dataset

In this paper, we use the CCPD dataset for training, which is a large, diverse, and carefully labeled open-source dataset of Chinese urban license plates (Xu et al., 2018). Data is available at https://github.com/detectRecog/CCPD. After opening the link, find CCPD (Chinese City Parking Dataset, ECCV), which can be downloaded directly through the Google Drive link. In order to conform to the license plate recognition problem studied in this paper, the dataset is processed in this paper. The target region of the license plate is cropped by the corner points of the license plate labeled by the CCPD dataset and corrected by an affine transformation.

Part of the license plate data is shown in Fig. 6. The CCPD dataset provides 100,000 images as a training set and 99,996 images as a validation set for the conventional scenario, and part of the images of the training set and validation set are shown in Fig. 6 (a).

In order to evaluate the recognition ability of the model in various extreme situations, the experiments use four important test sets of extreme situations in CCPD, which are Challenge, Blur, DB, and FN. Challenge is used to test the performance of the model in the case of challenging license plate images; Blur is used to test the performance of the model in the case of blurred license plate images; DB is used to test the model's performance in the case of too low or too high illumination; and FN is used to test the model's performance in the case of low resolution. Some of the images of the four test sets are shown in Fig. 6 (b). In addition, the CCPD dataset also provides a comprehensive test set, which is a comprehensive sampling of several extreme cases, and the images in this part are corrected by affine transformation.

In addition, in order to evaluate the spatial adaptation ability of the model, the experiments were evaluated using the Rotate and Tilt test sets in CCPD without affine transformation. As shown in Fig. 6 (c), rotate and Tilt test sets in CCPD with the test set without affine transformation. As shown in Fig. 6 (c), Rotate is used to test the performance of the license plate in the case of rotation; the license plate in Tilt is used to test the performance of the license plate in the case of extreme angles; and finally, the experiment is evaluated on the test set without affine transformation correction, which is used to evaluate the performance of the model without the use of corrective transformations.

Training parameter settings

The experiments were conducted using the processed CCPD training set with the training parameters shown in Table 1.

Assessment of indicators

The evaluation of the license plate recognition task should not only refer to whether the model is correct in recognizing characters, but also whether the model is correct in recognizing the number of characters. Therefore, TN1 is defined as the number of samples in which the number of characters is recognized incorrectly, TN2 is defined as the number of samples in which the characters are recognized incorrectly, and TP is defined as the number of samples in which the number of characters is recognized correctly and the characters are recognized correctly. The evaluation indexes are introduced: Length Consistency Rate (LCR) and Accuracy. The calculation formula is shown in Equations (10) and (11):

$$LCR = \frac{TN2 + TP}{TN1 + TN2 + TP} \tag{10}$$

$$Accuracy = \frac{TP}{TN1 + TN2 + TP} \tag{11}$$

The indicator LCR can reflect the accuracy of the number of characters recognized; the closer to 1 means that the number of characters recognized accurately, the greater the number of characters; the indicator Accuracy can reflect the number of characters and the accuracy of character recognition at the same time.

Experimental results and visualization

The LPRNet method based on graphically supervised signal-assisted training proposed in this paper is used to do side-by-side comparison experiments with models such as AlexNet (Krizhevsky Sutskever and Hinton, 2017), ResNet (He, et al., 2016), LPRNet, CRNN, and the experimental parameters are shown in Table 1. The accuracy of each model is evaluated on the validation set with different test sets, and the results are shown in Table 2. The code related to the experiments in this paper is publicly available and can be accessed via the following GitHub link: https://github.com/GENERjia/License-Plate-Recognition-Based-on-Graphically-Supervised-Signal-Assisted-Training. Or download it via the DOI link:https://doi.org/10.0.20.161/zenodo.14201550.

In order to compare the performance of different models, the computational volume, number of parameters, FPS, and average accuracy of different models are counted, as shown in Table 3. From Table 3, it can be seen that the proposed algorithm achieves the highest accuracy with the lowest number of parameters and computations. It proves the effectiveness of the proposed algorithm. The computational amount, number of parameters, and average accuracy of each model in Table 3 are visualized as shown in Fig. 7.

The comprehensive comparison of proposed algorithm with other common license plate recognition algorithms is shown in Fig. 8. Fig. 8(a) shows the comparison of the computational volume and accuracy between this paper and several common models; the x-axis indicates the computational volume of the model; the size of the diameter of the dots indicates the number of

model parameters; and the y-axis indicates the average accuracy in the test set. Fig. 8(b) shows the comparison of the accuracy of several common models on different test sets.
The experiments were recorded for the metrics during the training of CRNN, LPRNet, ResNet, and AlexNet, as shown in Fig. 9.
The results show that LPRNet is much higher than other models in both accuracy and convergence speed.
The experiment uses LPRNet as a Baseline and introduces graphically supervised signals on the basis of LPRNet. The experiment records the metrics during the training process of the LPRNet model and proposed algorithm, as shown in Fig. 10.
The results show that the accuracy of proposed algorithm is slightly higher than that of the original model on the validation set, and higher accuracies are obtained on all test sets, which proves that proposed algorithm can show good recognition results. In terms of the convergence speed and loss of the model, proposed algorithm is similar to the original model, i.e., this paper's algorithm does not need more iterations for optimization while improving the accuracy.
In order to more intuitively reflect the improved effect of the algorithm, different license plates from the experimental findings will be shown in the experiment. The results are shown in Fig. 11. For low light, blur, low resolution, angular offset, motion offset, etc., proposed algorithm can get more and more accurate detection results.
In order to further analyze the results, the model output can be visualized, as shown in Fig. 12(a). It is the classification output of LPRNet, and proposed algorithm is more accurate for the identification of blank characters and can distinguish between characters and the interval between them; Fig. 12(b) is the prediction output of proposed algorithm for the image label. The prediction output of the image label can basically realize the extraction of effective information from the image, and the prediction of the image label can basically react to the classification output situation.
In order to study the recognition of each character in the dataset, we conducted ablation experiments to better validate the effectiveness of proposed algorithm. And because of the uneven balance of the province character categories in the dataset, the experiments were only conducted on the recognition of numeric characters and alphabetic characters, as shown in Table 4.
Visualize the increase of Precision, Recall, and F1 Score indicators for each character using the proposed algorithm in the form of a bar chart, and analyze the increase using a histogram, as shown in Fig. 13. For the recognition of numeric characters, proposed algorithm not only enhances the precision rate of character recognition but also improves the recall rate, enabling it to recognize the characters that the original algorithm can not. As for the alphabetic characters, while proposed algorithm reduces the precision rate of the recognition of some characters, it can greatly improve the recall rate of character recognition so that the F1 Score can be improved.
For LPRNet and proposed algorithm, experiments were performed to visualize Grad-CAM (Selvaraju et al., 2017) on both models as shown in Fig. 14 in order to observe the regions of interest on the pictures for each category.

By observing the visualization results, it can be found that proposed algorithm can pay better attention to the correct position of each character without focusing too much on the other regions, which is especially obvious in the model's understanding of the "-" character, and proposed algorithm can better capture the segmentation and positioning relationship between characters. In order to evaluate the spatial adaptation ability of the model, the experiments are conducted to evaluate LPRNet and proposed algorithm in the Rotate, Tilt test set, and the test set without affine transformation respectively, and the results are shown in Table 5.
The results show that in the case of large angle change, both LPRNet and proposed algorithm can not achieve better results, and the effect of proposed algorithm is slightly lower than that of LPRNet. While in the case of no affine transformation correction, i.e., test set, proposed algorithm is slightly higher than that of LPRNet, which indicates that this paper's algorithm still achieves better results in the case of small angle change.

Conclusion

This paper presents a study on license plate recognition based on graphically supervised signal-assisted training and fully demonstrates the effectiveness and robustness of the algorithm. The biggest advantage of the proposed algorithm is that it can improve the model effect without modifying the neural network structure and increasing the extra computation. The average accuracy in complex cases is improved by 5.86%, Length Consistency Rate is improved by 12%, the average precision of characters is improved by 2.1%, and the average recall is improved by 6.9%. This shows that the research in this paper is able to improve the character recognition rate sufficiently to recognize the characters that were not recognized. The cost of improving the effect is that it needs to mine the graphical supervisory signals in the training data, and it also increases the training time of the model and the memory occupation of training. The training process proposed in this study is applicable to the vast majority of convolutional neural networks. The next step of this research can be applied to other image recognition tasks by introducing graphical supervisory signals, and even extended to non-textual image classification tasks. The key to exploring the problem lies in the method of mining graphical supervisory signals in regular non-textual images.

References

Ceng G.X., Ke X. 2021. 3D Convolution-Based Image Sequence Feature Extraction and Self-Attention for License Plate Recognition Method, Chinese Journal of Intelligent Science and Technology 3(3): 268-279.

Chen T., Kornblith S., Norouzi M., Hinton G., 2020. A Simple Framework for Contrastive Learning of Visual Representations, in: Proc. International Conference on Machine Learning, 2020.

Chen X., He K. (2021) Exploring Simple Siamese Representation Learning, in: Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

Devlin J., Chang M.W., Lee K., Toutanova K. 2018. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805.

505 Graves A., Fernández S., Gomez F., Schmidhuber J., 2006. Connectionist temporal classification:
506 labelling unsegmented sequence data with recurrent neural networks, in: Proc. 23rd
507 international conference on Machine learning, 2006.
508 He K., Chen X., Xie S., Li Y., Dollár P., Girshick R. 2022. Masked Autoencoders are Scalable
509 Vision Learners, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern
510 Recognition, 2022.
511 He K., Fan H., Wu Y., Xie S., Girshick R., Momentum Contrast for Unsupervised Visual
512 Representation Learning, in: Proc. 2020 IEEE/CVF Conference on Computer Vision and
513 Pattern Recognition (CVPR), 2020.
514 He K., Zhang X., Ren S., Sun J. 2016. Deep Residual Learning For Image Recognition, in: Proc.
515 IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.
516 Jin X., Tang R., Liu L., Wu J. 2021. Vehicle License Plate Recognition for Fog-Haze
517 Environments, IET Image Processing 15(6): 1273-1284.
518 Khan M.A., Sharif M., Javed M.Y., Akram T., Yasmin M., Saba T. 2018. License Number Plate
519 Recognition System Using Entropy-Based Features Selection Approach with SVM, IET
520 Image Processing 12(2): 200-209.
521 Kim D., Kim J., Park E. 2024. AFA-Net: Adaptive Feature Attention Network in Image
522 Deblurring and Super-Resolution for Improving License Plate Recognition, Computer
523 Vision and Image Understanding 238103879.
524 Krizhevsky A., Sutskever I., Hinton G.E. 2017. Imagenet Classification with Deep
525 Convolutional Neural Networks, Communications of the ACM 60(6) 84-90.
526 Lee Y., Lee J., Ahn H., Jeon M., 2019. SNIDER: Single Noisy Image Denoising and
527 Rectification for Improving License Plate Recognition, in: Proc. IEEE/CVF International
528 Conference on Computer Vision (ICCV), 2019.
529 Xu Z., Yang W., Meng A., Lu N., Huang H., Ying C., Huang L. 2018. Towards end-to-end
530 license plate detection and recognition: A large dataset and baseline, in: Proc. Proceedings
531 of the European conference on computer vision (ECCV), 2018.
532 Li X.P., Min W.D., Han Q., Liu R.K. 2019. License Plate Location and Recognition Based on
533 Deep Learning, Journal of Computer-Aided Design & Computer Graphics 31(6): 979-987.
534 Michelucci U. 2022. An introduction to Autoencoders, arXiv preprint arXiv:2201.03898.
535 Min D., Lim H., Gwak J. 2020. Improved Method of License Plate Detection and Recognition
536 Facilitated by Fast Super-Resolution Gan, Smart Media Journal 9(4): 134-143.
537 Pirgazi J., Pourhashem Kallehbasti M.M. 2022. Ghanbari Sorkhi A., An End-to-End Deep
538 Learning Approach for Plate Recognition in Intelligent Transportation Systems, Wireless
539 Communications and Mobile Computing 2022(1): 3364921.
540 Radford A., Narasimhan K., Salimans T., Sutskever I. 2018. Improving Language Understanding
541 by Generative Pre-Training, Preprint (2018).
542 Rao W.J., Gu Y.H., Zhu T.T., Huang Y.T. 2021. Intelligent License Plate Recognition Method in
543 Complex Environment, Journal of Chongqing Institute of Technology 35(3): 119-127.

Raza M.A., Qi C., Asif M.R., Khan M.A. 2020. An Adaptive Approach for Multi-National Vehicle License Plate Recognition Using Multi-Level Deep Features and Foreground Polarity Detection Model, Applied Sciences 10(6): 2165.

Selvaraju R.R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D. 2017. Grad-cam: Visual Explanations from deep Networks via Gradient-Based Localization, in: Proc. IEEE International Conference on Computer Vision, 618-626.

Shi B., Bai X., Yao C. 2016. An End-To-End Trainable Neural Network for Image-Based Sequence Recognition and its Application to scene Text Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 39(11): 2298-2304.

Wang H., Chen L. 2024. Deep Residual Network and Attention Mechanism for Special License Plate Recognition, Computer Engineering and Design 45(1): 291-298.

Wu C.H., Wu X.B., Wang L. 2019. Prospects for the Development of Intelligent Transportation Under the Background of a Strong Transportation Country, Journal of Transportation Research 5(4) 26-36.

Xu F., Chen C., Shang Z., Peng Y., Li X. 2024. A CRNN-based Method for Chinese Ship License Plate Recognition, IET Image Processing 18(2):298-311.

Xu H., Zhou X.D., Li Z., Liu L., Li C., Shi Y. 2021. EILPR: Toward End-To-End Irregular License Plate Recognition Based on Automatic Perspective Alignment, IEEE Transactions on Intelligent Transportation Systems 23(3): 2586-2595.

Yang Y., Wang J., Jiang J.L. 2024. Method of vehicle License Plate Recognition in Haze Weather Based on Deep Learning, Chinese Journal of Liquid Crystals and Displays 39(2): 205-216.

Yang Y., Xie J.N., Xu H.L. 2024. Developing Smart Transportation to Improve Travel Efficiency, People's Daily, April 2, 2024 (17).

Zhang C.Z., Li Y., Kang B.L., Chang Y. 2021. Blurred License Plate Character Recognition Algorithm Based on Deep Learning, Laser & Optoelectronics Progress 58(16): 251-258.

Zhang L., Wang P., Li H., Li Z., Shen C., Zhang Y. 2020. A Robust Attentional Framework for License Plate Recognition in the Wild, IEEE Transactions on Intelligent Transportation Systems 22(11) 6967-6976.

Zhang M., Liu W., Ma H. 2018. Joint License Plate Super-Resolution and Recognition in One Multi-Task Gan Framework, in: Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

Zhang M., Xie F., Zhao J., Sun R., Zhang L., Zhang Y. 2018. Chinese License Plates Recognition Method Based on a Robust and Efficient Feature Extraction and BPNN Algorithm, Journal of Physics: Conference Series 1004: 12022.

Zherzdev S., Gruzdev A. 2018. Lprnet: License Plate Recognition via Deep Neural Networks, arXiv preprint arXiv:1806.10447.

Zhu S., Wang Y., Wang Z. 2024. A Lightweight License Plate Detection Algorithm Based on Deep Learning, IET Image Processing 18(2): 403-411.

Table 1(on next page)

Neural network training parameters

1 **Table 1.** Neural network training parameters.

Parameter	Value
Epoch	10
Image size	(94, 24)
Batch size	256
Learning rate	1.00E-03
optimizer	Adam
Weight decay	2.00E-05
Dropout rate	0.5

2

3

Table 2(on next page)

Comparison of the accuracy of scene license plate recognition models on each CCPD test set

Table 2. Comparison of the accuracy of scene license plate recognition models on each CCPD test set.

DataSet	Metrics	CRNN [6]	LPRNet [7]	AlexNet [33]	ResNet-18[34]	Ours (Compared to LPRNet)
val	LCR	0.9981	0.9992	0.9954	0.9959	**0.9996** (+0.0004)
	Accuracy	0.9903	0.9964	0.9861	0.9845	**0.9972** (+0.0008)
test	LCR	0.7212	0.7278	0.6503	**0.8629**	0.8538 (+0.1259)
	Accuracy	0.4715	0.5739	0.4508	0.5523	**0.6370** (+0.0631)
blur	LCR	0.6446	0.5219	0.4365	**0.8473**	0.7954 (+0.2735)
	Accuracy	0.3386	0.3468	0.2626	**0.4580**	0.4558 (+0.1090)
challenge	LCR	0.7640	0.7089	0.6697	**0.8880**	0.8461 (+0.1371)
	Accuracy	0.4921	0.5386	0.4602	0.5865	**0.6022** (+0.0637)
db	LCR	0.5764	0.5473	0.4252	0.7700	**0.7769** (+0.2296)
	Accuracy	0.3291	0.4053	0.2782	0.3861	**0.5144** (+0.1091)
fn	LCR	0.7548	0.7755	0.7199	0.8609	**0.8681** (+0.0927)
	Accuracy	0.5320	0.6461	0.5166	0.5904	**0.6963** (+0.0501)
rotate	LCR	0.7835	0.8861	0.8242	0.9121	**0.9190** (+0.0329)
	Accuracy	0.6419	0.8236	0.6792	0.7142	**0.8413** (+0.0177)
tilt	LCR	0.7067	0.8081	0.7329	0.8482	**0.8939** (+0.0857)
	Accuracy	0.4768	0.6916	0.4995	0.5351	**0.7470** (+0.0554)
Average	LCR	0.7437	0.7468	0.6818	**0.8732**	0.8691 (+0.1222)
	Accuracy	0.5340	0.6278	0.5166	0.6009	**0.6864** (+0.0586)

5

Table 3(on next page)

Comparison of FLOPs, number of parameters, FPS on CPU and average accuracy among models

Table 3. Comparison of FLOPs, number of parameters, FPS on CPU and average accuracy among models.

Models	FLOPs	Params	FPS	Accuracy
CRNN[6]	0.95 G	7.3 M	92.30	0.5340
LPRNet[7]	0.29 G	0.4 M	93.24	0.6278
AlexNet[33]	0.63 G	3.8 M	90.03	0.5166
ResNet-18[34]	1.09 G	11.2 M	76.39	0.6009
Ours	0.29 G	0.4 M	93.24	0.6864

4

Table 4(on next page)

Recognition of numeric and alphabetic characters

1 **Table 4.** Recognition of numeric and alphabetic characters.

	LPRNet[7]			Ours		
Char	Precision	Recall	F1-score	Precision	Recall	F1-score
0	0.814	0.743	0.777	0.87	0.819	0.844
1	0.795	0.771	0.783	0.837	0.87	0.853
2	0.817	0.782	0.799	0.871	0.886	0.879
3	0.858	0.779	0.817	0.913	0.852	0.882
4	0.796	0.775	0.785	0.82	0.862	0.841
5	0.803	0.784	0.794	0.877	0.87	0.874
6	0.855	0.766	0.808	0.903	0.858	0.88
7	0.828	0.768	0.797	0.894	0.86	0.877
8	0.839	0.765	0.8	0.893	0.832	0.861
9	0.819	0.797	0.808	0.902	0.874	0.888
A	0.947	0.955	0.951	0.92	0.971	0.945
B	0.919	0.679	0.781	0.904	0.74	0.814
C	0.893	0.795	0.841	0.875	0.875	0.875
D	0.9	0.639	0.747	0.855	0.72	0.781
E	0.889	0.771	0.826	0.907	0.817	0.86
F	0.871	0.763	0.813	0.876	0.858	0.867
G	0.863	0.743	0.799	0.905	0.78	0.838
H	0.926	0.701	0.798	0.932	0.773	0.845
J	0.87	0.782	0.824	0.852	0.873	0.862
K	0.915	0.733	0.814	0.909	0.822	0.863
L	0.926	0.686	0.788	0.907	0.814	0.858
M	0.905	0.764	0.828	0.904	0.847	0.874
N	0.89	0.717	0.794	0.897	0.801	0.847
P	0.904	0.775	0.835	0.935	0.833	0.881
Q	0.808	0.814	0.811	0.811	0.831	0.821
R	0.909	0.757	0.826	0.944	0.8	0.866
S	0.801	0.78	0.791	0.884	0.806	0.843
T	0.875	0.711	0.785	0.895	0.747	0.814
U	0.851	0.708	0.773	0.831	0.795	0.813
V	0.813	0.768	0.79	0.849	0.811	0.83
W	0.718	0.801	0.757	0.786	0.86	0.822
X	0.869	0.782	0.823	0.863	0.861	0.862
Y	0.858	0.791	0.823	0.842	0.856	0.849
Z	0.889	0.731	0.803	0.901	0.764	0.826
Average	0.860	0.761	0.806	0.881	0.830	0.854

2

3

Table 5(on next page)

Accuracy of the model when the image undergoes a change in spatial angle

Table 5. Accuracy of the model when the image undergoes a change in spatial angle.

DataSet (no Affine)	LPRNet[7]	Ours	increase
rotate	0.0017	0.0006	-0.11%
tilt	0.0020	0.0010	-0.10%
test	0.1946	0.2110	1.64%

Figure 1

LPRNet network structure diagram

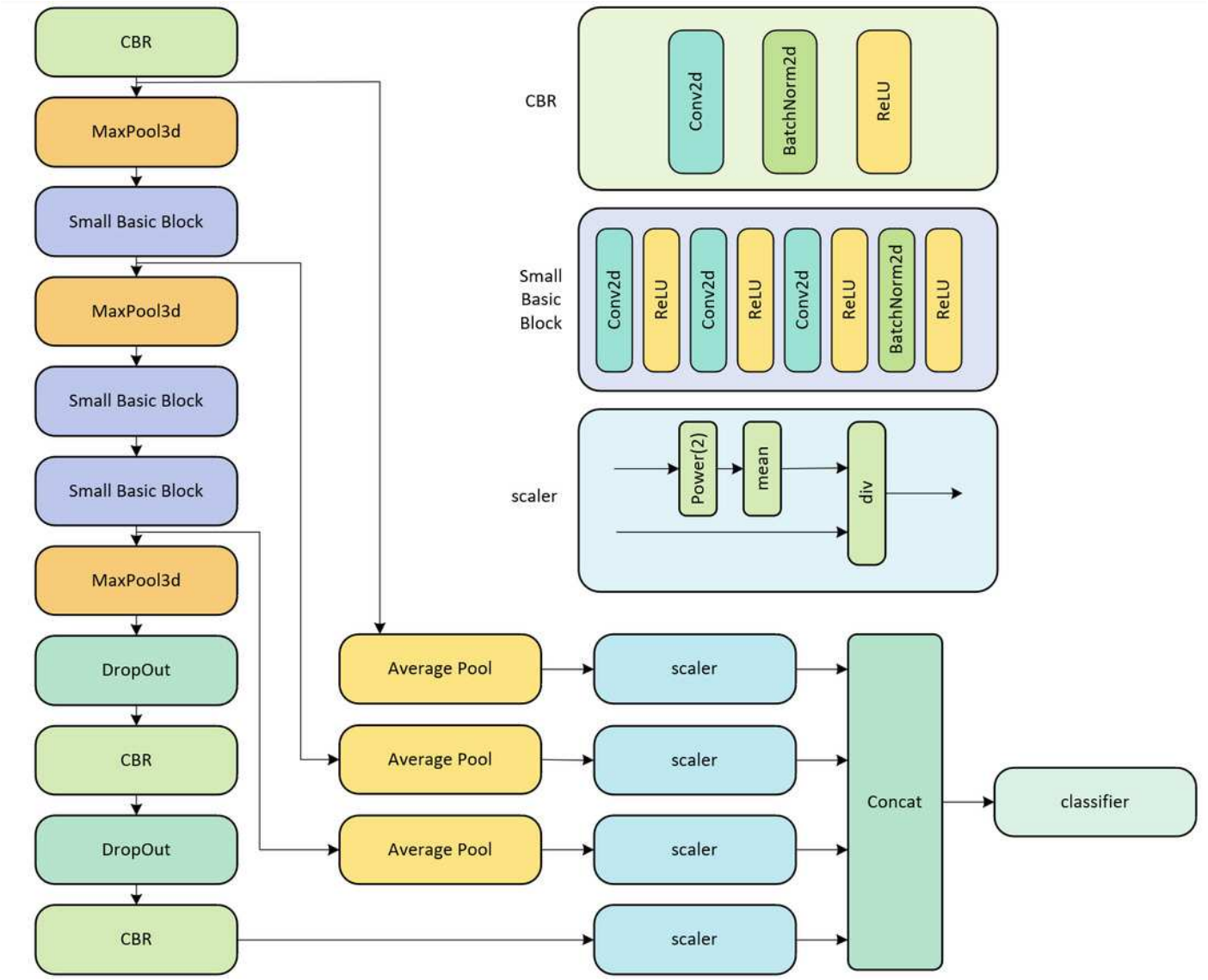

Figure 2

Two representations of string labels

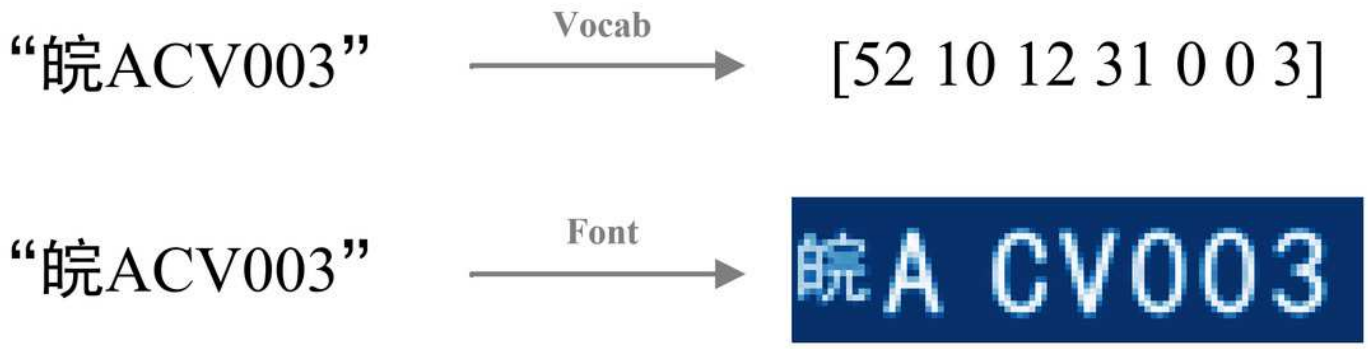

Figure 3

Spatial alignment of image label

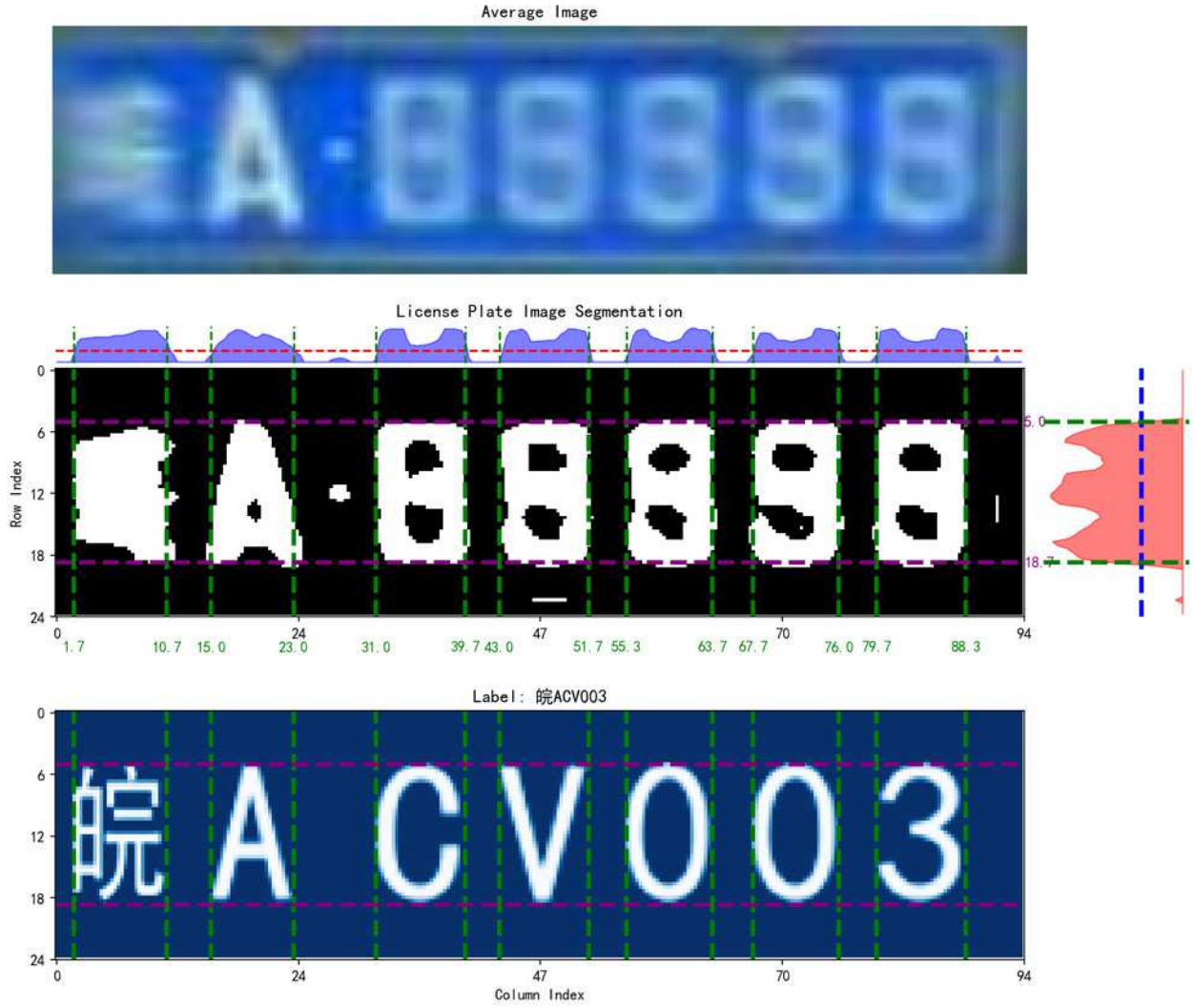

Figure 4

Position of the upsampling decoder and schematic of its structure

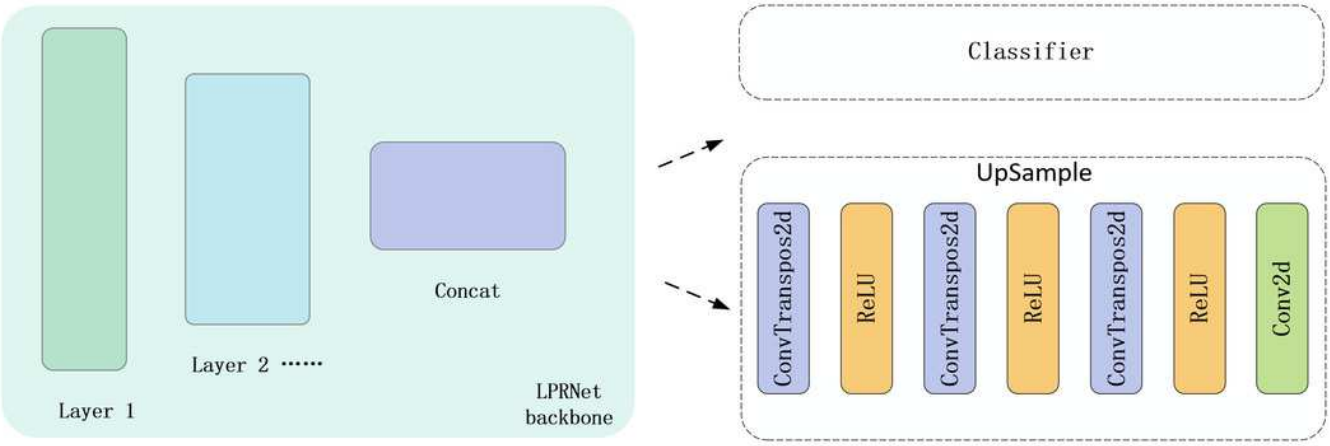

Figure 5

Flow of algorithm for graphical supervised signal-assisted training

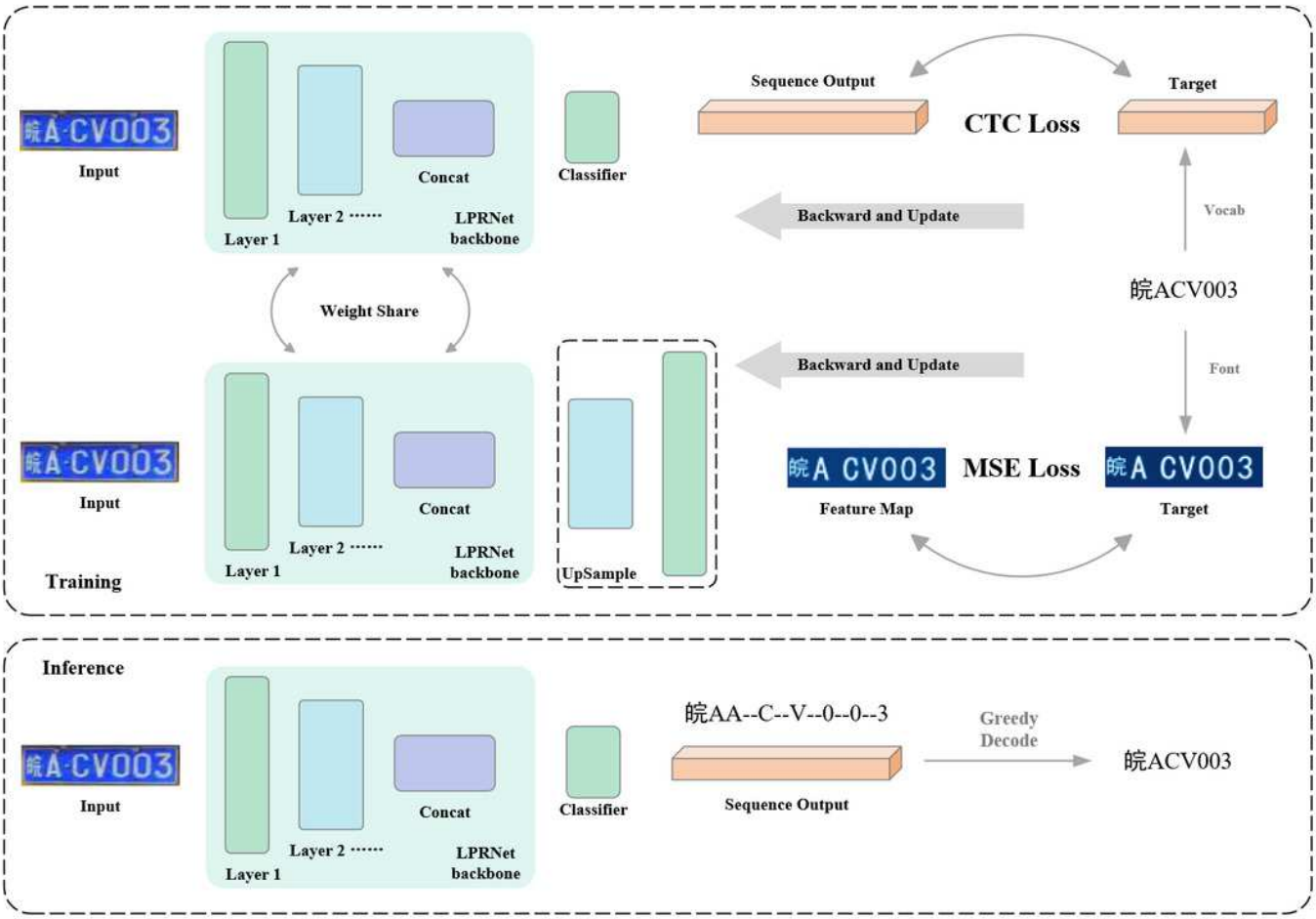

Figure 6

Example of CCPD dataset

(a) Train set and Validation set; (b) Challenge set, Blur set, DB set, FN set; (c) Rotate set, Tilt set, No Affine Transformation set

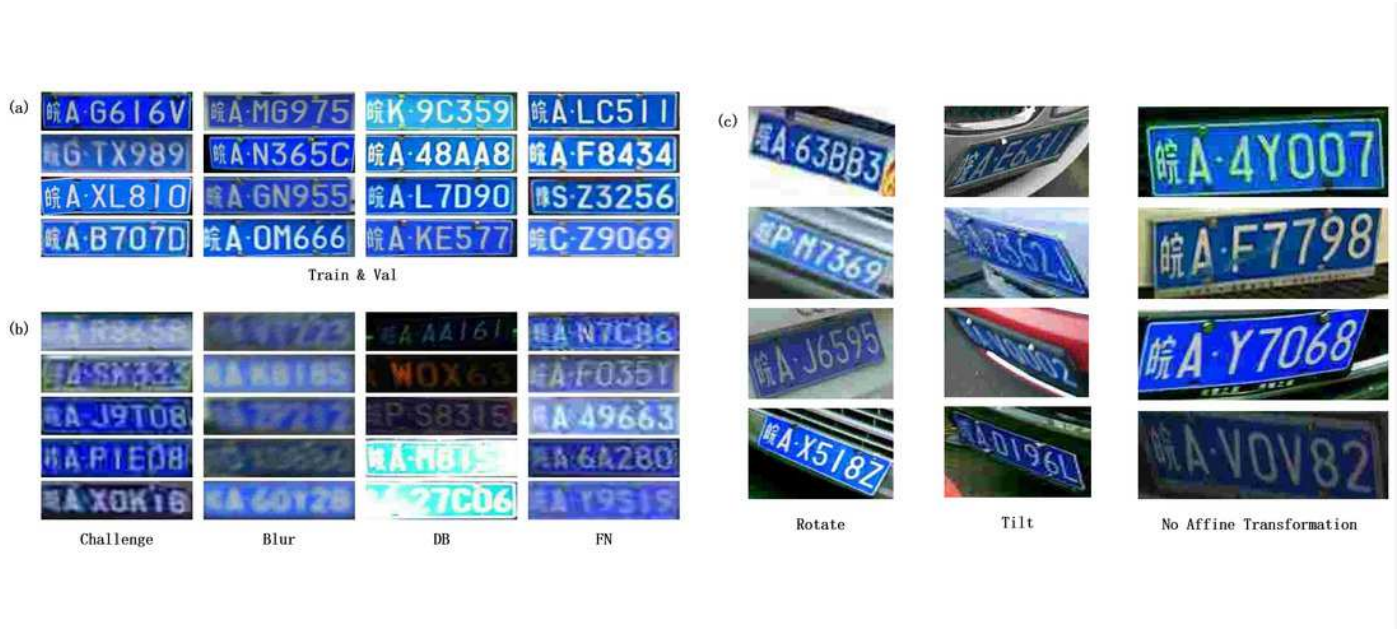

Figure 7

Visualization of models comparison

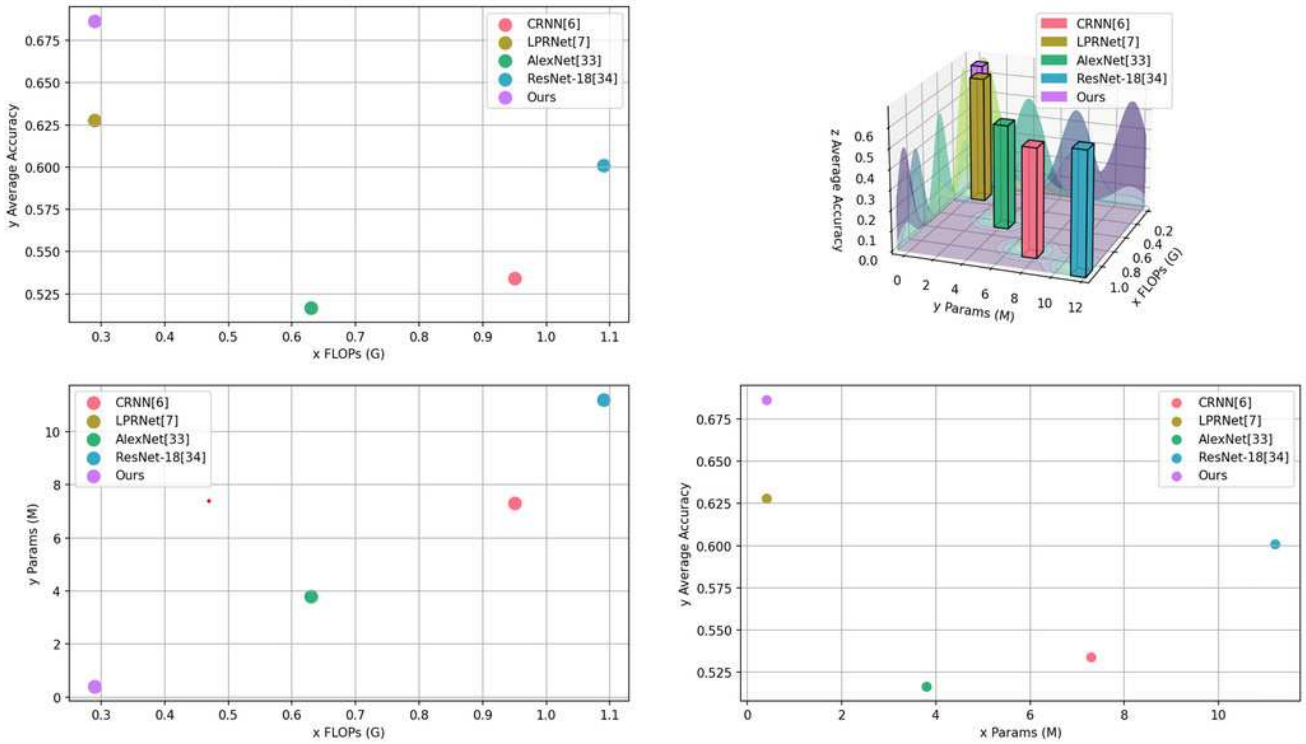

Figure 8

Comparison of this paper's algorithm with other common algorithms

(a) Model Accuracy vs. Computational Cost; (b) Comparison of Models on Different Datasets

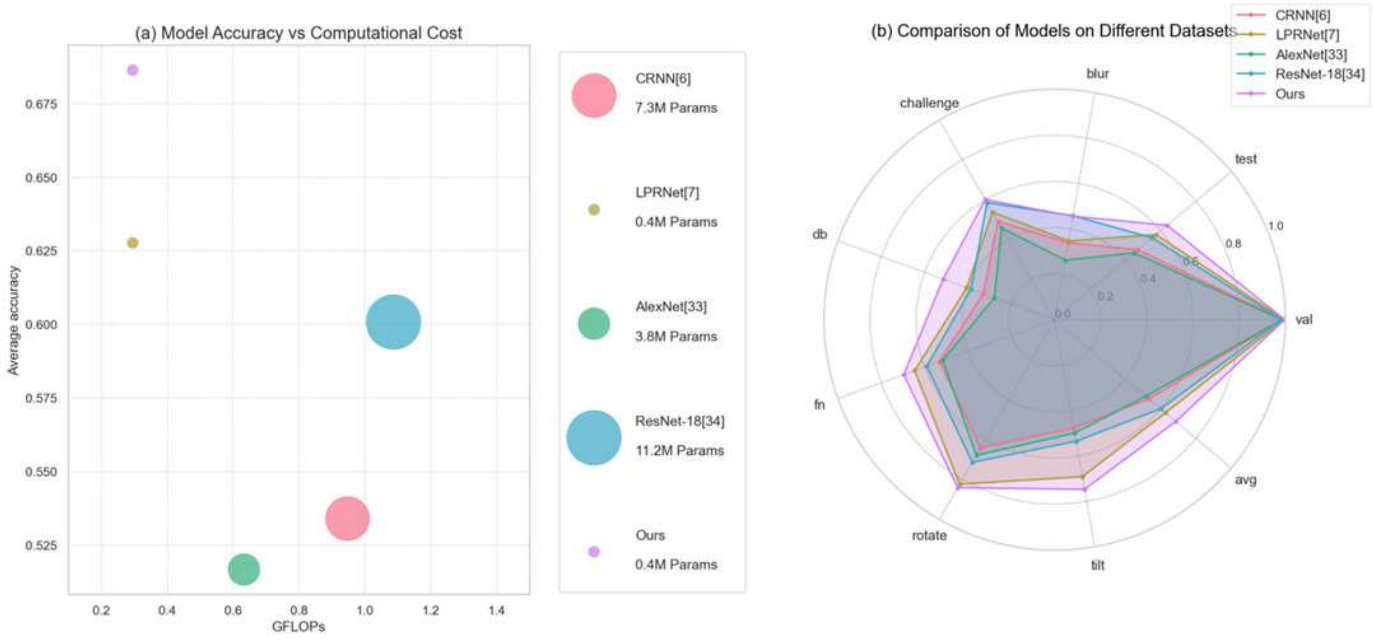

Figure 9

Comparison of Losses and Accuracy in the Validation Set of the Training Process Model

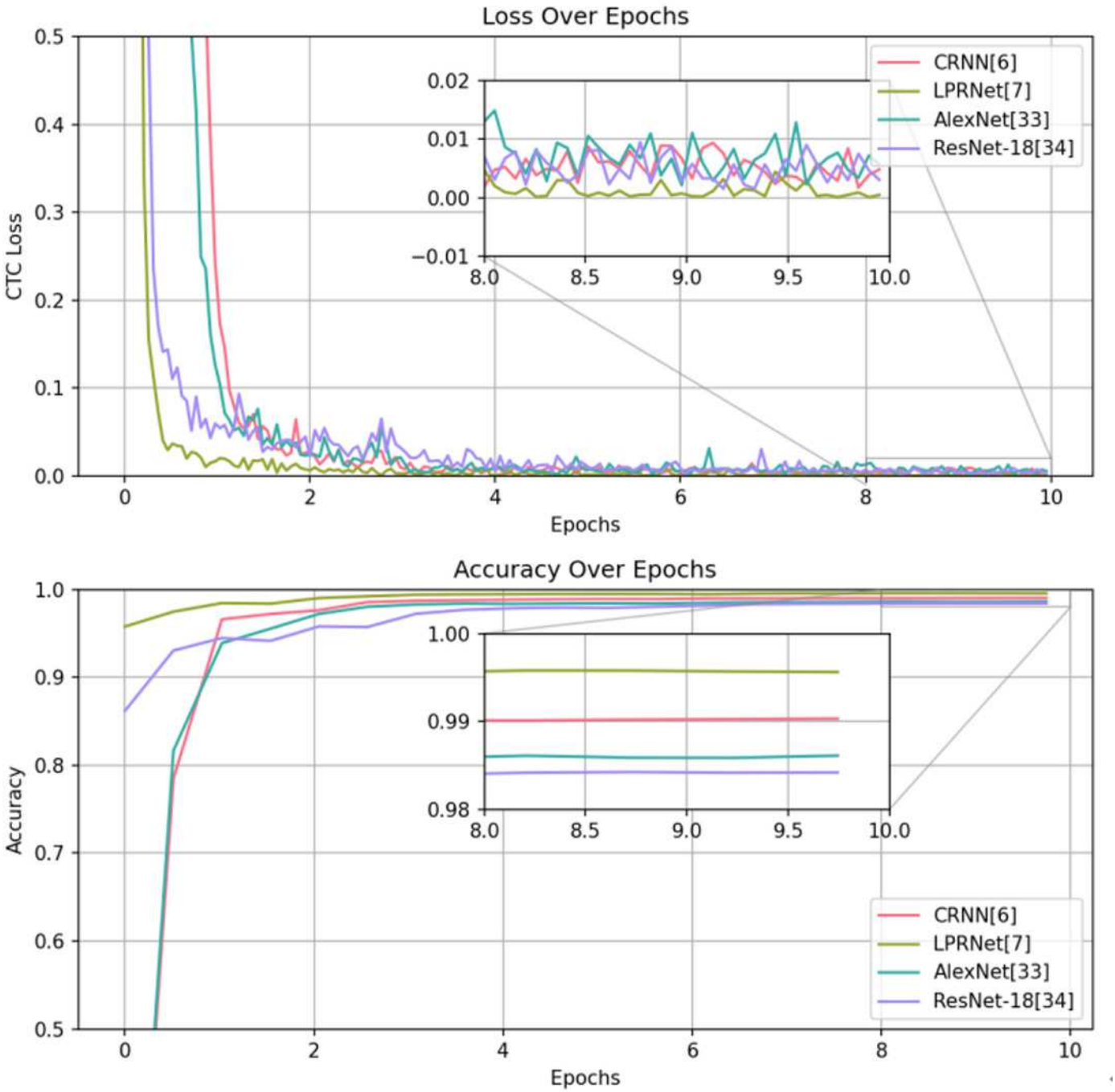

Figure 10

Comparison of accuracy and loss between LPRNet and LPRNet validation set based on graphically supervised signal-assisted training during the training process

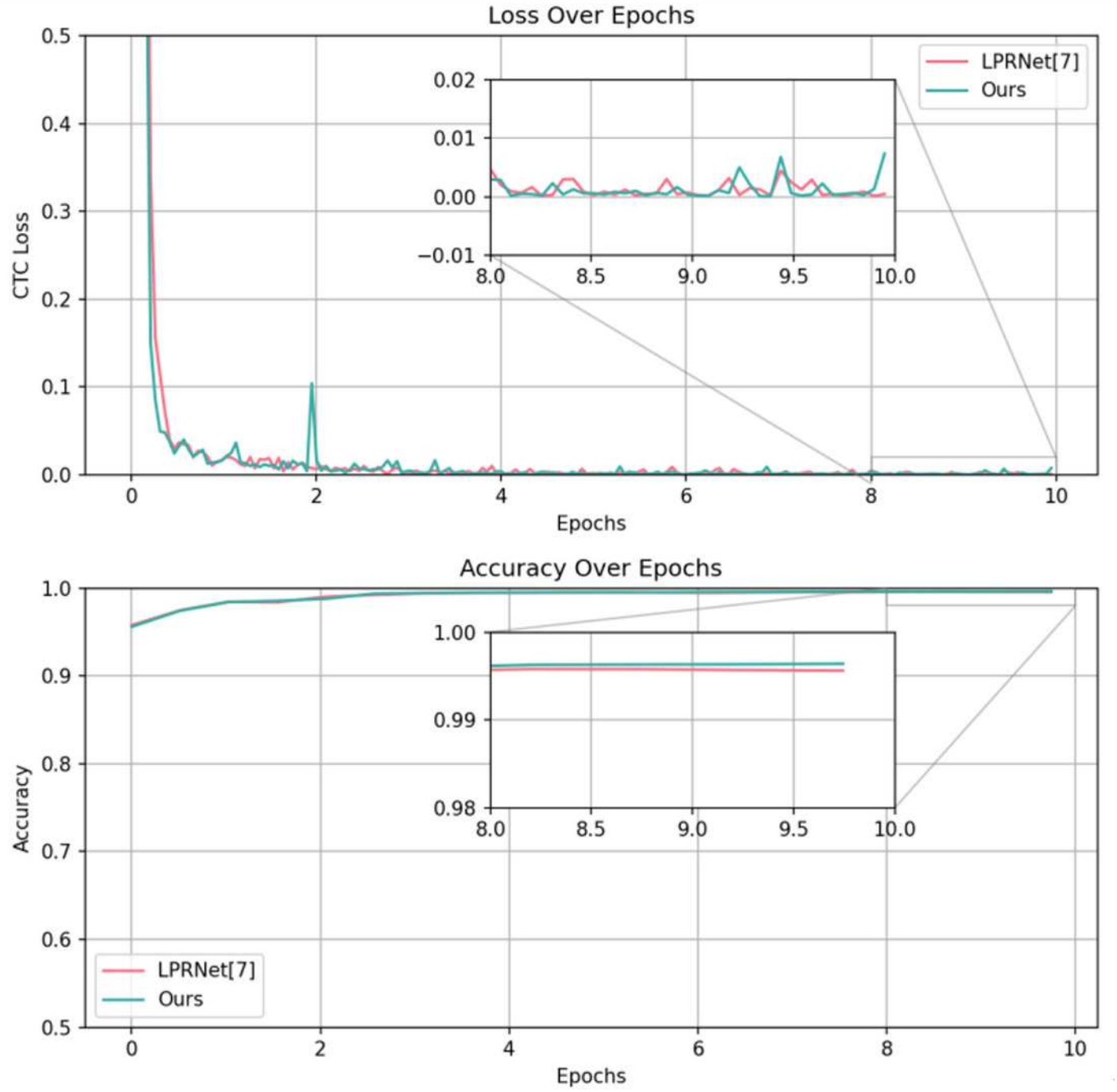

Figure 11

Recognition effect of different models on license plate in different cases

Figure 12

Model output results

(a) Classification output; (b) Prediction output of proposed algorithm for the image label

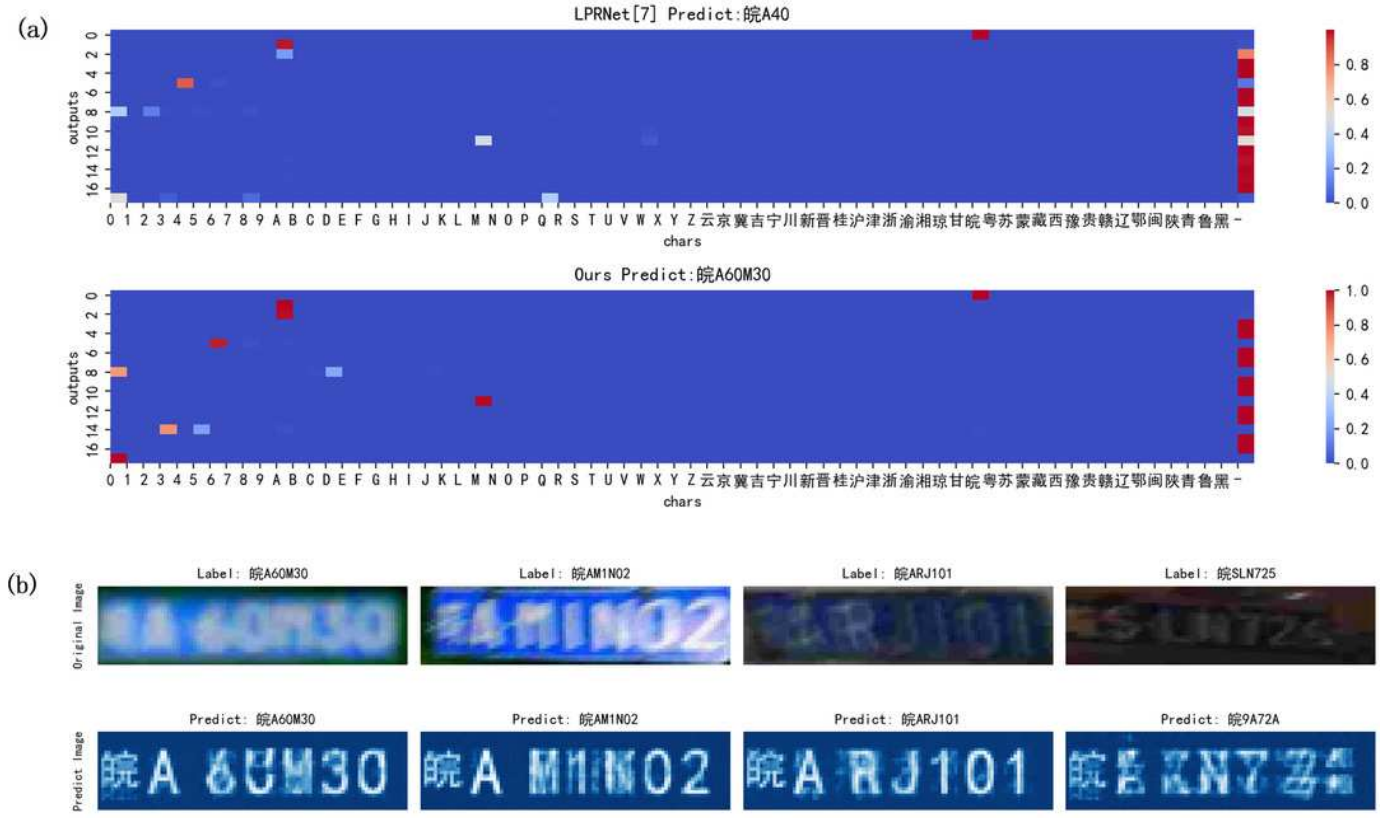

Figure 13

Increase in Precision, Recall, and F1 Score metrics for each character

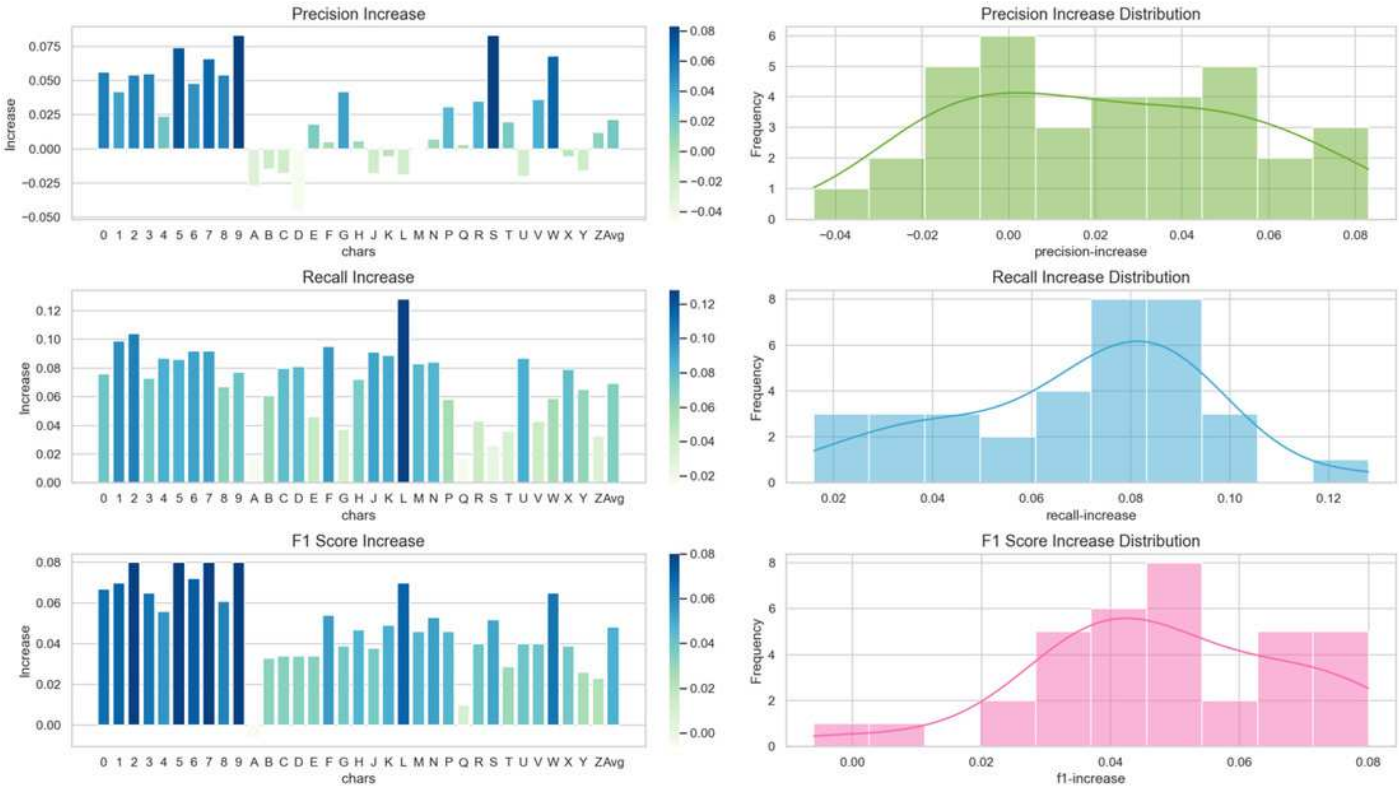

Figure 14

Comparison of Grad-CAM visualization for different models