J. Vis. Commun. Image R. 104 (2024) 104314

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Efficient license plate recognition in unconstrained scenarios☆,☆☆

Chao Wei [c], Fei Han [b], Zizhu Fan [a,*], Linrui Shi [d], Cheng Peng [e]

[a] School of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, 201306, China
[b] School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China
[c] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China
[d] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China
[e] Department of Engineering, King's College London, Strand, London WC2R 2LS, United Kingdom

ARTICLE INFO

Keywords:
License plate detection
License plate recognition
Efficient detection
Deep learning

ABSTRACT

Automatic license plate recognition (ALPR) is a critical technology for intelligent transportation systems. Most existing ALPR methods are focused on specific application scenarios. Although there are a few methods that focus on unconstrained scenarios, they are very time-consuming. In this work, we propose an efficient ALPR (EALPR) framework, where we can handle distorted license plates (LP) caused by perspective problems with high efficiency. We design a light LPD structure based on efficient object detection methods and use anchor-free strategies for LPD to alleviate the problem of expensive costs. Benefitting from these optimizations and a united framework structure, the proposed EALPR has real-time efficiency. We evaluate our method on five datasets and the results show that our method achieves state-of-the-art accuracy: 98.15% on OpenALPR(EU), 95.61% on OpenALPR(BR), 99.51% on AOLP(RP), 88.81% on SSIG, 79.41% on CD-HARD. Additionally, our method achieves an impressive speed of 74.9 FPS (Frames Per Second), outperforming existing approaches and demonstrating its efficiency. Our source code can be accessed at https://github.com/wechao18/Efficient-alpr-unconstrained.

1. Introduction

In recent years, the traffic congestion in the urban area has become more serious than before due to the rapid increase of vehicles. The congestion often leads to many traffic problems such as traffic jams, traffic accidents, and traffic offenses. Automatic License Plate Recognition (ALPR) is one of the critical technologies to solve the problem of traffic congestion effectively and can improve traffic management efficiency. It has been widely used in traffic monitoring and successfully used in some restriction scenarios such as parking tolls and other applications. Besides, ALPR has also been applied in unconstrained complex scenarios such as intelligent monitoring and location tracking. However, existing ALPR methods are not applicable to the above high-computing scenarios, they are not accurate enough or too inefficient. It is of great significance to develop an efficient ALPR algorithm for unconstrained scenarios.

Most ALPR methodologies can be decomposed into two primary components: license plate detection (LPD) and license plate recognition (LPR). As we know, the current challenges of ALPR are focused on the first part (i.e. LPD), so existing ALPR methods are devoted to designing a superior license plate detector [1–3]. In the early period, traditional image processing methods [4–7] are usually used to deal with LPR. These methods used low-level features of an image such as edges, colors, and textures to solve the problem of LPR in simple scenes where the license plate images are in nearly ideal condition. That is, the license plate takes up a large part of the image, and it is not tilted in the image which is high-quality [8–10]. In this case, the traditional methods can achieve desirable performance. However, if license plates are within unconstrained and complex conditions such as image rotation, license plate occlusion and blurring, and large illumination variants, license plate detection may become very challenging under these unconstrained scenarios. It is possible that the unconstrained scenarios can lead to degrading the detection performance significantly. Thus, it is relatively difficult to design a robust method based on the low-level features under the unconstrained scene. To address this problem, many methods of license plate detection are

☆ This paper has been recommended for acceptance by Zicheng Liu.

☆☆ This research was supported in part by the Natural Science Foundation of China, grant number 61991401, 61976108 and Jiangxi Provincial Natural Science Foundation of China, grant number 20192ACBL20010.

* Corresponding author.
E-mail addresses: wechao@tongji.edu.cn (C. Wei), hanfei@ujs.edu.cn (F. Han), zzfan3@163.com (Z. Fan), shilinrui180@qq.com (L. Shi), cheng.2.peng@kcl.ac.uk (C. Peng).

https://doi.org/10.1016/j.jvcir.2024.104314
Received 18 June 2023; Received in revised form 19 September 2024; Accepted 13 October 2024
Available online 19 October 2024

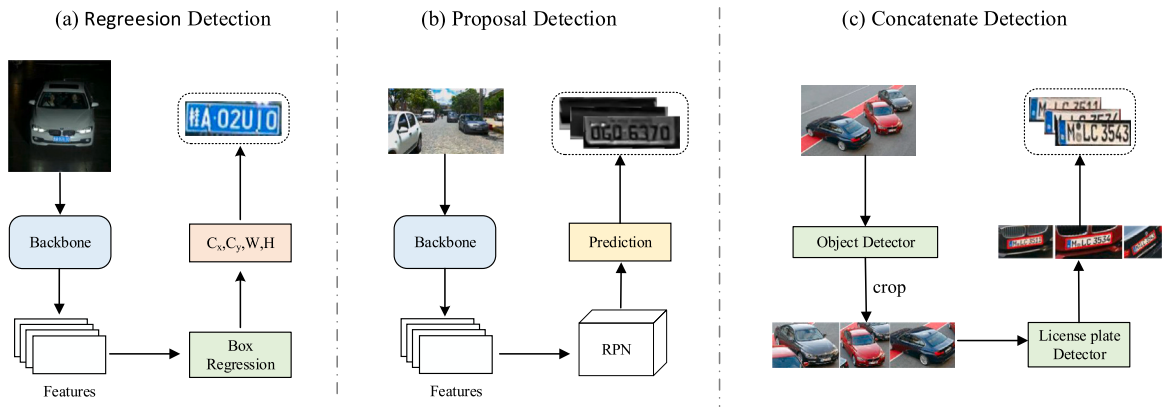

Fig. 1. Previous CNN-based LPD framework types. Previous CNN-based LPD frameworks can be broadly categorized into (a) regression detection framework that predicts one license plate location directly from features. (b) proposal detection framework that proposes all potential license plate locations and then predicts separately. (c) concatenate detection framework that detects vehicles to narrow the search and then predicts the license plate in each vehicle image.

developed based on the Convolutional Neural Network (CNN) which automatically learns more powerful feature representations from large amounts of image data. CNN-based methods have been widely applied in various challenging recognition tasks [11–13], and some LPR methods focus on dealing with complex scenarios exploited by CNN-based algorithms [1,14,15]. Although these methods are improved by the object detection framework and perform better than traditional methods, most of them are still valid only under specific conditions or strong assumptions. Specifically, we summarize the current common CNN-based methods depending on their framework as follows (as in Fig. 1): (i) regression detection where features are extracted via a deep network and then predict center coordinates and size of a license plate through a fully connected network [16]; (ii) proposal detection where consider LPD as an object detection task and using existing object detection frameworks [1]; (iii) concatenate detection where two independent deep networks are used to detect vehicles and license plates, respectively [17,18]. While the first framework is fast, it cannot handle the case where there are multiple vehicles in a single image. The second framework does not suffer from the above-mentioned defects, but it is easy to fail for long-distance license plate detection. The last framework does not encounter the problems of the former two methods and constitutes the state-of-the-art methods in ALPR [18]. However, due to the file I/O operations between the two networks, the overall high latency is caused even if the networks are lightweight.

In this work, we propose an efficient ALPR framework called EALPR. LPD is the most challenging step of ALPR in unconstrained scenes. The typical LPD based on CNN is often time-consuming and the efficiency of ALPR is strongly dependent on the LPD. Aiming to improve the efficiency of license plate detection (LPD), we propose an efficient LPD structure inspired by CenterNet [19] and EfficientDet [20]. The proposed LPD structure detects vehicle position and makes full use of existing feature maps, avoiding both the extra I/O overhead and the difficulty of direct LP detection, as shown in Fig. 2. Therefore, it has a good balance between performance and time consumption. We also evaluate the performance of ALPR on a level playing field and achieve performance that is competitive with the state-of-the-art ALPR methods. The main contributions of our work are summarized as follows:

(1) We alleviate the problem of the expensive costs by sharing features between two independent networks and propose an applicable training strategy to ensure feature diversity.

(2) We propose a general EALPR framework that can be generalized to multiple modern object detection architectures.

(3) The performance of our EALPR framework reaches the forefront of the field across various datasets, e.g. a 98.15% LPR accuracy (1.3% higher than SOTA) on OpenALPR(EU), 95.61% LPR accuracy (0.9% higher than SOTA) on OpenALPR(BR), 99.51% LPR accuracy (0.3% higher than SOTA) on AOLP(RP), 88.81% LPR accuracy (0.3% higher than SOTA) on SSIG and 79.41% LPR accuracy (same as SOTA) on CD-HARD. Moreover, we achieved 74.9 FPS (Frames Per Second) running speed, far more than the latest methods.

The remainder of this paper is organized as follows. In Section 2, we briefly summarize ALPR. Section 3 describes the proposed method in detail. In Section 4, we show the performance comparison on five datasets and study some necessary ablation experiments. Finally, we make conclusions in Section 5.

2. Related work

The present section delineates contemporary ALPR methodologies founded upon deep learning. The task of ALPR can be divided into two successive phases: license plate detection (LPD) and license plate recognition (LPR). We review their related work next.

2.1. License plate detection

In recent years, deep learning-based methods have yielded very impressive results in some object detection tasks [21–25]. Inspired by these methods, many researchers have gradually expanded these methods to LPD tasks [1,17,26] because LPD is similar to object detection. The proposed LPD method in [1] based on YOLO [24] is typical, where the authors achieve a high detection accuracy by improving the original framework. They consider the angle and the image area proportion of the LPs and use two CNNs for cascade prediction. The prepositive network roughly predicts a license plate as an attention region, which can remove redundant information. The attention region is input in the latter network to predict a precise rotational rectangular region. However, the angle prediction, unable to adapt the LPs distortion in real scenarios. Also, they lack an LPR process to evaluate the performance of a complete ALPR. A similar method proposed by Silva et al. [26] recycled a single CNN to complete the cascade prediction, in which the authors evaluate the performance of LPR.

We observe that the recent YOLO-based networks [23,25] achieve better computational efficiency and classification performance than the previous detection framework. There are many LPD methods based on the improvement of such networks [17,18]. This indicates that it is a feasible way to use the existing structural design to perform license plate detection, which can significantly reduce a lot of computational costs by using a pre-trained model. In addition to the detection-based methods, there are other studies like [16] that directly regress LPs location parameters, in which the inefficient sliding window process is skipped via an end-to-end regression network. In [16], the authors detect the coordinates of a license plate via a full convolution network. Although such methods are highly efficient, they focus on the LP images

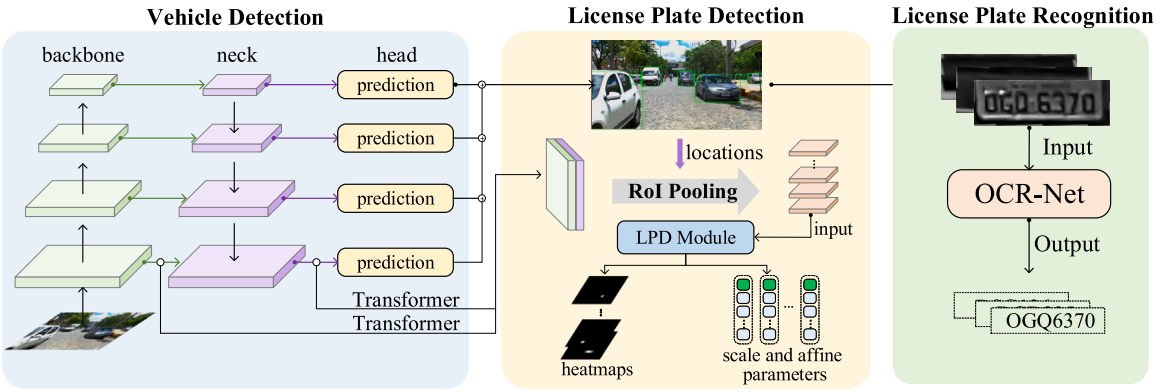

Fig. 2. The structure of our EALPR framework. The object detector adopts a standard for vehicle detection and feature extraction. The transformer encoder module is added for more efficient context feature extraction.

taken at close range and are only applicable to the case of a single license plate in an image.

Recently, some anchor-free object detection methods have been proposed [19,27]. This type of method has advantages in speed, because of its reduced computation cost compared with anchor-based detection methods. These methods usually return the heatmaps of the object's key points. In the heatmaps, the area corresponding to the positive sample and its surroundings is activated, which greatly reduces the computational complexity without anchor regression. Fortunately, only a small number of objects can match the default anchor boxes in the special task of license plate detection, which is not conducive to the regression of the bounding box during the training process. Therefore, the model design in this task borrowed the anchor-free idea as an effective way to reduce the computation cost without significantly degrading the detection accuracy.

2.2. License plate recognition

License plate recognition (LPR) is the last step of the ALPR system. This task can be seen as simplified OCR, which has been studied for a long time. LPR methods can be divided into two categories: segmentation-based methods and segmentation-free methods.

Segmentation-based methods are the most common and have achieved impressive performance. Such methods usually use detection-based methods to separate out each character in the LP. For examples, the LPR models in [28–30] are modified by YOLO [24,25]. Their work all benefits from current work in object detection and is highly flexible. However, they cannot handle the distorted characters.

For segmentation-free methods, in [15,31], LPR was considered as a sequence labeling problem that does not require annotating the localization. It extracts a sequence of features from the LP region by CNN. Additionally, it amalgamates Recurrent Neural Network (RNN) architecture with Connectionist Temporal Classification (CTC) to annotate the sequential data, employing CTC for plate decoding devoid of character segmentation. Compared to the methods in [15,31], the structures proposed by [16,32] are more simplified and identify characters by multiple convolutional branches based on the number of characters.

In order to overcome the shortcomings of the previous works, we propose an EALPR framework. In this framework, we design a light LPD structure based on the trained object detection method. Our method achieves state-of-the-art performance on multiple datasets.

3. The proposed method

The overall structure of our EALPR framework is shown in Fig. 2. The proposed method contains two processes: license plate detection (LPD), and license plate recognition (LPR). Given an input image $I \in R^{C\times H\times W}$, license plates (LPs) in the image will efficiently detect and then rectify them to a front projection view. Subsequently, the rectified license plates undergo recognition by the LPR module, extracting characters and arranging them accordingly. This section will first present the design of the LPD, then introduce the LPR module and characters post-processing rules, and finally detail the framework architecture.

3.1. License Plate Detection (LPD)

Many methods [26,33,34] adopt an end-to-end pipeline for LPD, predicting the center of each license plate. However, it is hard to predict them in a wide range of scenarios because the vehicle may cover a small percentage of the image. Considering the LPD capability in various scenarios, we first detect vehicles to obtain the regions of interest (RoI), which constrains the search area of the LPD process, as shown in Fig. 2. As we know, various vehicles exist in widely used detection datasets, such as ImageNet [35], Pascal-VOC [36], and COCO [37]. Therefore, it is very reasonable to exploit the models that were trained in these datasets to complete the vehicle detection process.

For the LPD, we design a light module to solve the location of license plates. The module takes the pooled small-size features as input, greatly improving the efficiency of detection. The LPD module is a fully convolutional network that detects a single object (object uniqueness in an RoI), as shown in Fig. 4. It encodes a set of affine transformation parameters to warp the rectangle into a parallelogram that recovers the distortion of LP on the vehicle. So, we can convert a distorted LP to a front-parallel projection, and simplify the problem of LPR.

3.1.1. Network architecture

The overall architecture is shown in Fig. 2. We employ standard FPN detection frameworks [19,20,23,27] to provide features. The maximum size features from the backbone and neck are used to detect the license plates in an image. These features are pooled to get multiple 28×28 RoI branches by pooling based on vehicle detection results. We used a more calibrated RoIAlign [38] method for the pooling process. Following the methodology of CenterNet [19], we forecast a proposal heatmap to delineate the distinctive license plate on a vehicle. Consequently, each branch predicts a proposal heatmap of shape $1\times H_p\times W_p$. Simultaneously, a parameter map of shape $C_p \times H_p \times W_p$ that contains the scale and affine parameters is predicted. For the license plate center point with the heatmap located at (x_p, y_p), the corresponding bounding box scale and affine parameters are contained in the C_p dimensional at (x_p, y_p).

All the convolutional layers employ 3×3 kernel size and 1 padding size. Batch normalization (BN) and Leaky Linear Units (LeakyReLU) are connected after each convolutional layer and form a convolution block. The module consists of a sequence of convolution blocks, except for the final layer that is responsible for the localization. For the final prediction layers, the proposal heatmaps are activated by a sigmoid

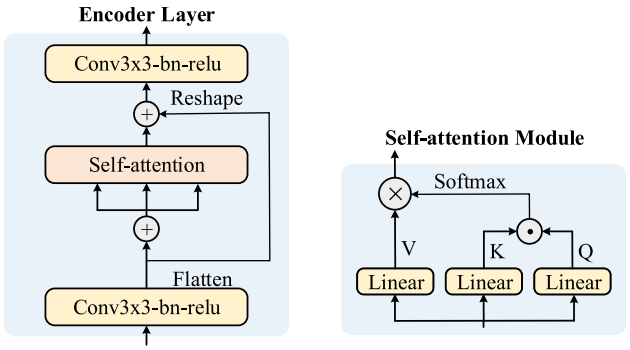

Fig. 3. The structure of the transformer encoder. The ⊕, ⊙, and ⊗ respectively represent matrix addition, dot-product operation and element-wise product operation.

function to obtain the probability of the object, and other parameter prediction layers are linearly activated. The details of the LPD module are shown in Fig. 4 and Fig. 5. We also add a transformer encoder to our framework. For example, the position of the LP depends on the features of the entire vehicle which has different categories and perspectives. Therefore, we exploit the attention mechanism to fuse contextual information. The structure of the transformer encoder used in our framework is shown in Fig. 3.

3.1.2. Loss function

Designing a proper loss function is very important for the model, which affects the performance of the model during the inference. For the LPD stage, a pre-trained model is used for vehicle detection. The loss function does not need to be considered in preliminary detection. Thus, we focus on the loss function of the LPD module, which can be divided into three parts. The first part of the loss function represents the probability error between the center of the LP and the prediction. The remaining two parts of the loss function are related to the LP location and the affine transformation parameters regression.

For the LP center location, each center point of the LP has a corresponding positive coordinate in a heatmap, and the other coordinate points are negative samples. However, such a setting will lead to a serious imbalance between positive and negative samples. Hence, we adopt focal loss [39] to constrain the proposal heatmap following CornerNet [27] and CenterNet [19]. Meanwhile, the penalty for negative samples near the center point of the LP is reduced. It uses a two-dimensional Gaussian distribution to define the penalty coefficient, set the radius parameter to 2, and generate a heatmap through the Gaussian kernel function $Y_{xy} = exp(-\frac{x^2+y^2}{2\sigma^2})$, where σ is 1/3 of the radius. The loss function is expressed as follows.

$$\ell_{heat} = \sum_{xy} \begin{cases} (1-\hat{Y}_{xy})^{\alpha} \log(\hat{Y}_{xy}) & Y_{xy} = 1 \\ (1-\hat{Y}_{xy})^{\beta} (\hat{Y}_{xy})^{\alpha} \log(1-\hat{Y}_{xy}) & otherwise \end{cases} \tag{1}$$

where Y_{xy} is the truth label at coordinate (x, y) and $\hat{Y}_{xy}$ represents the predicted value of the heatmap at coordinate (x, y), α and β represent the hyper-parameters of focal loss, which are set to 2 and 4 respectively in our experiments.

The center point location defined in Eq. (1) is not exact (It is rounded to ensure that it falls on the heatmap grid). Thus, we add an offset regression loss to get the exact location. For an LP center point (x, y) in an image, the position in a feature map after downsampling is $(\frac{x}{2^n}, \frac{y}{2^n})$, where n denotes the number of downsampling. The position offset is represented as follows.

$$\delta_{xy} = (\frac{x}{2^n} - \left\lfloor \frac{x}{2^n} \right\rfloor, \frac{y}{2^n} - \left\lfloor \frac{y}{2^n} \right\rfloor) \tag{2}$$

where δ_{xy} represents truth offset of the LP's center point (x, y), and $\lfloor\ \rfloor$ is a function to rounds down. The offset is limited to 0–1 so that each key point will not move to the area of other grid cells. We use SmoothL1-loss [21] for error calculation, and only positive samples are involved during training.

$$\ell_{off} = \sum_{i} \begin{cases} 0.5(\delta^i_{xy} - \hat{\delta}^i_{xy})^2 & |\delta^i_{xy} - \hat{\delta}^i_{xy}| < 1 \\ |\delta^i_{xy} - \hat{\delta}^i_{xy}| - 0.5 & otherwise \end{cases} \tag{3}$$

where δ^i_{xy} is the ith element in δ_{xy} and $\hat{\delta}^i_{xy}$ is the ith element in the predicted offset of the positive sample.

Similarly, we can also define a loss function for the size error. But we have a trick here, we found that if the LPs cannot be completely detected, it may seriously affect the LPR process. Therefore, we set different loss penalties for two cases and give a larger penalty for the case where the predicted bounding box is smaller than the truth bounding box. We use L1-loss to constrain the size as follows.

$$\ell_{scale} = \sum_{i} \begin{cases} |\varphi^i_{xy} - \hat{\varphi}^i_{xy}| & \hat{\varphi}^i_{xy} \geq \varphi^i_{xy} \\ \lambda |\varphi^i_{xy} - \hat{\varphi}^i_{xy}| & \hat{\varphi}^i_{xy} \leq \varphi^i_{xy} \end{cases} \tag{4}$$

where $\varphi_{xy} = (\frac{w}{2^n}, \frac{h}{2^n})$ denotes truth size of the LP in prediction layers and φ^i_{xy} is the ith element in it, $\hat{\varphi}_{xy}$ denotes the predicted size of the LP and $\hat{\varphi}^i_{xy}$ is the ith element in it, λ is a penalty coefficient which is set to 2 in our experiments.

The last part of the loss function represents the error of the affine transformation. According to the previously predicted bounding box information, the position of the four corners of the license plate can be calculated using affine parameters. Let $q_1 = [-\hat{w}, -\hat{h}]^T, q_2 = [\hat{w}, -\hat{h}]^T, q_3 = [\hat{w}, \hat{h}]^T, q_4 = [-w, h]^T$ respectively denote the four vertices of the bounding box. Let $p_i = [x_i, y_i]^T, (i = 1, \ldots, 4)$ denote the four corners of the LP. For each bounding box, there are six affine values to be predicted, which are used to correct the bounding box to match the distorted LP. Inspired by [17,18], we label the six values as a_1 to a_6. The affine transformation is defined as follows:

$$T_i(q) = \begin{bmatrix} max(a_1, 0) & a_2 & a_3 \\ a_4 & max(a_5, 0) & a_6 \end{bmatrix} \begin{bmatrix} q \\ 1 \end{bmatrix}^T \tag{5}$$

where the main diagonal parameter of the affine matrix is set to a non-negative number to avoid excessive rotation, so we use the max function on the a_1 and a_5 parameters.

$$\ell_{point} = \sum_{i=1}^{4} |T_i(q) - S_i| \tag{6}$$

where S_i represents the ground-truth value of the ith license plate vertex.

The total loss in the training phase is defined as follows.

$$\ell_{total} = \alpha \ell_{heat} + \gamma \ell_{off} + \beta \ell_{scale} + \eta \ell_{point} \tag{7}$$

where the coefficients α, β, γ and η in our experiment are simply set to 1.0, 0.2, 1.0, and 1.0 respectively for consistent order of magnitude.

3.1.3. Training details

We use an annotated dataset containing 196 images to train the proposed LPD model. We had to use ad hoc data augmentation strategies to cope with the small dataset. Considering that the model needs to handle diverse variations of LPs in real-world scenarios (perspective distortion, color, and illumination changes), we improve the model's robustness through a series of data augmentation strategies during training. Consequently, during training, geometric distortion, cropping, flipping, and photometric distortions (mimicking color and illumination variations) must be randomly applied to training images. To simulate the distortion in real scenarios, we artificially apply a random perspective transformation. Also, since our method uses region features as the input of the LPD module instead of cropping the original image [17,18]. Therefore, our augmentation strategy needs to be applied to the entire image while the methods proposed by [17,18] enhance the cropped image. The augmentation operations are all similar, but the implementation methods are different.

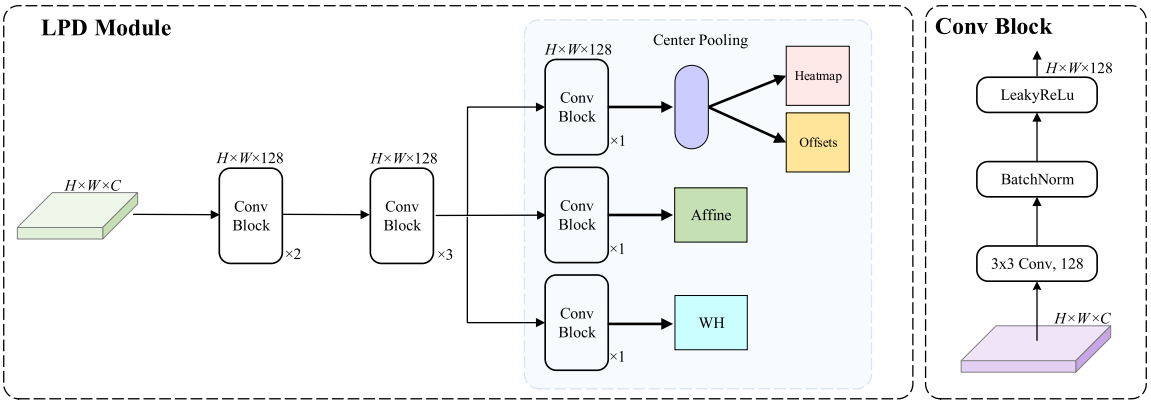

Fig. 4. License plate detection module. It employs the pooling method in [19]. Features of each branch are fed to the pooling layer before point prediction. It predicts the center point and LP size.

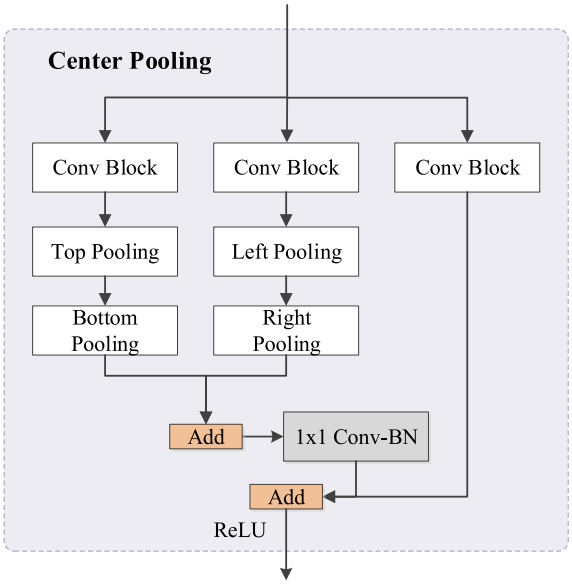

Fig. 5. Pooling architecture.
Source: The pooling strategy from [19].

Assuming a vehicle width $w \in [w_{min}, w_{max}]$, height $h \in [h_{min}, h_{max}]$, and the corresponding LP width $w' \in [w'_{min}, w'_{max}]$, height $h' \in [h'_{min}, h'_{max}]$. We first randomly crop the original image, and in which we simultaneously modify the vehicle annotation. The crop operation is performed randomly but is constrained to preserve the main body area of the vehicle. We can get multiple input features by compressing the range of the initial bounding box (no need to expand the range because no area outside the vehicle will be detected). The compressing range is limited to the LP boundary. So, the horizontal compress is bound to $h_c \in [0, min(w'_{min} - w_{min}, w_{max} - w'_{max})]$ and the vertical compress is bound to $v_c \in [0, min(h'_{min} - h_{min}, h_{max} - h'_{max})]$.

Then we rotate the cropped image via random perspective transformation. We set the roll, pitch, and yaw angles range values at ±40°, ±40°, ±25°, respectively. These angles provide a very wide variability perspective, and a planar homography H generated based on these parameters is used to perform a perspective transformation on the cropped image. The perspective change matrix H Ref. [17,18], which contains the same camera focal length and plane distance. We warp the cropped image according to H (the vehicle bounding box is rectified after warped).

Finally, we change color and random flip with 0.5 probability. Based on the above strategies, we can extend a small-scale dataset as a way

Fig. 6. Different augmentations for the same sample. The green bounding box and cyan quadrilateral represent the augmentation process of the vehicle region and the LP region, respectively.

Table 1
LPR comparison of different methods on each dataset.

	OpenALPR		AOLP	SSIG	CD-HARD	FPS
	EU	BR	RP			
EALPR	**98.15**	**95.61**	**99.51**	**88.81**	**79.41**	**74.9**
IWPOD-NET^{-} [18]	94.44	94.74	98.36	–	79.41	26.8
	Literature					
IWPOD-NET [18]	97.22	94.74	98.36	–	82.35	26.8
WPOD-NET [17]	93.52	91.23	98.36	88.56	75.00	15.6
Laroca et al. [30]	96.9	–	99.2	–	–	–
Li et al. [31]	–	–	88.38		–	–
Li et al. [34]	–	–	83.63	–	–	–
Hsu et al. [33]	–	–	85.70[a]	–	–	–
Bjorklund et al. [40]	–	–	96.9	–	–	–
Selmi et al. [41]	–	–	96.3	–	–	–
	Commercial systems					
OpenALPR	88.89	97.37	93.62	87.44	59.62	–
Sighthound	94.44	98.25	94.93	81.46	72.12	–

[a] In [33], the authors provided an estimative and not a real evaluation.

Table 2
Evaluation datasets.

Database	Vehicle Dist.	Images
OpenALPR(EU)	Close	104
OpenALPR(BR)	Close	108
AOLP(RP)	Close	804
SSIG(test)	Medium, far	611
CD-HARD	Close, medium, far	102

to improve the generalizability of the model. We show each step of the data augmentation in Fig. 6.

3.2. License Plate Recognition (LPR)

The LPs detected by the LPD process are rectified, so the problem of LPR is greatly simplified. Based on the above conditions, there are several approaches available for LPR modules, most of which are discussed in Section 2. In our method, we use trained OCR-net presented in [17] for final evaluation. It requires a 80×240 image as input and trained with large datasets [30].

Our dataset comes from multiple regions, and although the arrangement rules for license plates vary from region to region. For example, the AOLP(RP) license plate comprises two parts. One part invariably consists of four digits, while the other part, containing two characters, must include at least one letter. The characters on the LPs of OpenALPR (BR) and SSIG follow the sequence of three letters followed by four numbers. In OpenALPR(EU) and CD-HARD, although license plates originate from different regions, they all consist of both numerical and alphabetical segments. There are some conflicts between letters and numbers, such as $B \leftrightarrow 8$, $Z \leftrightarrow 2$, $I \leftrightarrow 1$, $S \leftrightarrow 5$, etc. If the above numbers appears where the letters should appear, then it can be replaced with a letter, and vice versa. Therefore, during our testing, we can streamline the process without concerning ourselves with specific arrangement rules. It is important to note that our replacement strategy assumes that the type of license plate is known. We use the existing OCR network for recognition that does not contain the type of license plate identification, so it does not classify the area where the license plate is located.

4. Experiments

4.1. Experimental setting

4.1.1. Datasets

To reflect the extensiveness of the proposed method, we conduct experiments on four datasets: OpenALPR, AOLP [33], SSIG [26], and Cars [42]. The OpenALPR dataset is provided by a commercial company for benchmark tests that support the OpenALPR library. The SSIG dataset contains 2000 high-resolution images from 101 different cars, in which the vehicles are far away from the camera. Due to their high resolution, they need to be compressed before detection, which leads to the size of the LP being too small for detection. Therefore, it is currently a challenging dataset. For street camera scenes, the AOLP (RP) subset is a challenging dataset in the case of deformation, which attempts to simulate the circumstances where the camera is installed in a patrolling vehicle or held by a person. The CD-HARD dataset is selected from the Cars dataset [42] and manually marked with 102 challenging images. The selected images are all readable by humans and the license plate has excessive deformation. The details of the datasets are shown in Table 2.

4.1.2. Evaluation metrics

Although the final LPR results can be used as an evaluation metric for ALPR systems, we still additionally present the results of the LPD module to evaluate a full pipeline of ALPR systems. For evaluating LPD, we adopt the Intersection over Union(IoU) between the predicted LP and the GT label, which is a standard metric in object detection. As mentioned in method [18], LPD may be misleading. We present in Section 4.2 a brief analysis of our LPD module.

For LPR results, the accuracy is the number of correctly recognized LPs divided by the numbers of all test datasets (all characters and order in the LP need to be recognized correctly). We present LPR results in Section 4.3.

4.1.3. Implementation details

The proposed method contains three detection processes, we empirically set their confidence thresholds to 0.3 (vehicle detection), 0.3 (LP detection), and 0.4 (character detection), respectively. In the LPD module, the weights of the vehicle detection network are trained on the COCO dataset, while the LPD network weights are randomly initialized. During training and testing, the input images are resized to the input resolution required by the detector, e.g. 640×640 (YOLOv5), 416×416 (YOLOv3). In the optimizing process, we use Adam optimizer [43] and step learning rate decay [44] with an initial learning rate of 3e−4. We train 10 000 epochs on the training set with a batchsize of 8. The results are reported on the OpenALPR, AOLP, SSIG, and CD-HARD. We use an efficient detection network YOLOv5s for LPD and OCR-Net [17] for the LPR phase. All the experiments were computed on a machine with an RTX3080 GPU.

Table 3
LPD accuracy comparison on AOLP (%).

Method	Subset					
	AC		LE		RP	
	PR	RE	PR	RE	PR	RE
Hsu et al. [33]	91	96	91	95	91	94
Li et al. [31]	98.5	98.4	97.8	97.6	95.3	95.6
T. Björklund et al. [40]	**100**	99.3	**99.8**	99.0	**99.8**	99.0
IWPOD-NET [18]	99.9	99.3	96.3	99.7	99.7	**100**
Ours	**100**	**99.6**	99.2	**99.9**	**99.8**	**100**

Table 4
LPD accuracy comparison on CCPD (%).

Method	Subset						
	DB	FN	Rot.	Tilt	Weath.	Chall.	Tot.
RPNet [16]	89.5	85.3	94.7	93.2	84.1	92.8	89.30
IWPOD-NET [18]	86.1	84.3	94.8	93.0	95.7	**93.4**	89.71
Ours	**90.2**	**90.8**	**96.9**	**95.1**	**98.6**	92.9	**94.08**

4.2. License plate detection results

We simply adopt the AOLP datasets for LPD evaluation (focus only on the results of the LPD is not significant, as mentioned in [18]). Following [33] we use the bounding box annotations of the AOLP and evaluate the results in terms of precision (PE) and recall (RE). In the experiment, the condition for detection correct is if the IoU with a labeled bounding box is greater than 0.5. The results shown in Table 3 demonstrate that the proposed LPD method presents the best precision and recall rates (except the LE subset). It is crucial to emphasize that our approach can accurately detect certain license plates that are absent from the ground truth data, thereby elevating the count of false positives (and consequently, precision), notably within the LE subset.

Additionally, we perform an LPD evaluation on the CCPD dataset. For CCPD datasets, we set the IOU threshold to 0.7 for a fair comparison. The results shown in Table 4 demonstrate that our method can achieve better detection results except for the Challenge subset of CCPD.

4.3. License plate recognition results

Table 1 compares our method with other LPR methods on each dataset. It shows that our method achieves state-of-the-art performance on all test datasets. On the AOLP(PR) subset, compared to the best method, the accuracy of our method is improved by nearly 1.1%. On the OpenALPR(EU) and OpenALPR(BR) subsets, compared to the best method, the accuracy of our method is improved by nearly 3.7% and 0.9% respectively. On the SSIG dataset, we also have a little improvement, even if our method is not advantageous in the images with small LP (compared to the method of cropping the original images such [17,18]). Although the accuracy is the same on the CD-HARD dataset, our method achieves higher efficiency than other methods. In terms of efficiency, the EALPR framework achieves a throughput of 74.9 FPS throughout the entire process employing YOLOv5s. Even under a fair comparison using YOLOv3, we still achieved an outstanding result of 61.3 FPS, which far surpassing the 26.8 FPS of the method in [18], demonstrating the efficiency of our approach. It is important to note that IWPOD-NET^{-} in [18] is a fair comparison with our proposed method (IWPOD-NET has an extra post-processing method). Even so, we still outperform the IWPOD-NET. We give the training loss and testing accuracy with epochs curves as shown in Fig. 7.

To figure out the effect of each component, we perform ablation experiments on the test datasets. First, we replace the vehicle detection method with other commonly used methods, as shown in Table 5. We use four popular object detection methods YOLOv3, YOLOv5, YOLOv7, and EfficientDet for the ablation study. We can observe that using a more accurate detection method is beneficial in improving the ALPR performance. In addition, we found that YOLOv3 has lower accuracy compared to other models and [18]. This is due to our method being designed to enhance efficiency, leading to a partial loss of information during the ROI process. Simultaneously, YOLOv3 utilizes smaller image inputs compared to YOLOv5, resulting in smaller feature maps during the pooling process. This disadvantage is particularly evident in datasets with a small proportion of vehicles, such as SSIG. However, the speed based on YOLOv3 still surpasses the approach [18] by a considerable margin. Fortunately, this issue has been addressed in higher versions of YOLOv5, demonstrating that our framework design achieves higher efficiency while maintaining accuracy.

Table 5
Ablation study of the vehicle detection methods.

	OpenALPR		AOLP	SSIG	CD-HARD
	EU	BR	RP		
YOLOv3	96.30	95.37	98.36	85.13	74.51
EfficientDet	97.8	**95.61**	99.51	88.61	**79.41**
YOLOv5s	**98.15**	**95.61**	**99.51**	**88.81**	**79.41**
YOLOv7	**99.07**	**95.61**	**99.84**	**88.93**	**79.41**

Table 6
Ablation study of different RoI pooling methods.

	OpenALPR		AOLP	SSIG	CD-HARD
	EU	BR	RP		
RoIPool	97.62	95.3	99.20	68.96	64.71
RoIAlign	**98.15**	**95.61**	**99.51**	**88.81**	**79.41**

Table 7
Ablation study of different weighting strategies.

	OpenALPR		AOLP	SSIG	CD-HARD
	EU	BR	RP		
Standard	96.30	93.58	98.28	80.68	74.51
Strategic	**98.15**	**95.61**	**99.51**	**88.81**	**79.41**

We further analyze the pooling method which greatly improved detection performance in [38]. We use two methods from [21,38] for comparison, as shown in Table 6. In the experiment, we use the same vehicle detection method YOLOv5s for feature extraction. The results show that for the smaller LPs, such as the SSIG dataset, the improvement is nearly 20%. For the CD-HARD dataset that requires more precise positioning, the accuracy has also been improved by nearly 15%. In other datasets, the improvement is not obvious. These results suggest that the calibration pool layer is crucial for our model and facilitates training with a small object.

Table 7 shows the accuracy comparison under weighting strategy ablation in Eq. (4), where we adopt YOLOv5s detection and RoIAlign pooling method. The first row represents the results using the same weights. In the second row, the weighting strategy in Eq. (4) is used. We observe that the accuracy on the SSIG dataset has been significantly improved, and the accuracies on other datasets are also improved by 1%. The results also demonstrate that the weighting strategy for scale prediction is critical for our method.

Finally, we analyze the function of the transformer encoder. We compare the ALPR accuracy on test datasets, as shown in 8, in which all the configurations are the best. In the second row, the transformer function is used. The results demonstrate that the transformer is mainly beneficial to the LPs with a small percentage of the image. This suggests that LPs with a small percentage of the image are more dependent on contextual features and global information.

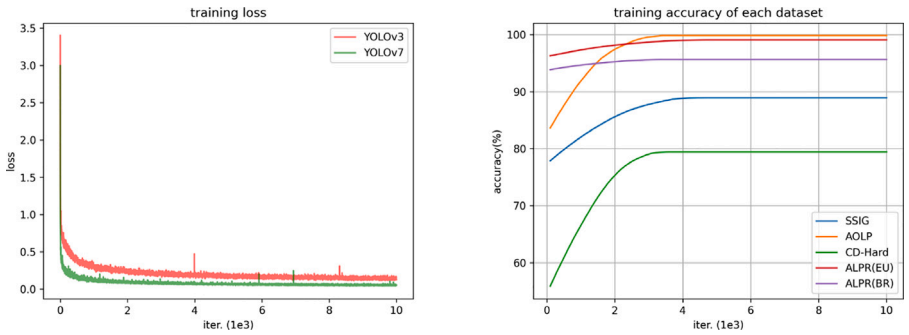

Fig. 7. Training loss and accuracy curves.

Table 8
Ablation study of the transformer.

	OpenALPR		AOLP	SSIG	CD-HARD
	EU	BR	RP		
Standard	**98.15**	**95.61**	99.32	84.21	**79.41**
S. w/o encoder	**98.15**	**95.61**	**99.51**	**88.81**	**79.41**

5. Conclusion

In this paper, we have designed an efficient ALPR framework for unconstrained scenarios. The proposed framework can efficiently detect the license plate because it uses shared region features and light network architecture instead of cropping the image to extract depth information. The experimental results demonstrate that the proposed method achieves state-of-the-art accuracy. However, there are still some problems to be solved for robust license plate detection. First, the lack of fully labeled license plate data from unconstrained scenarios limits the performance of our method, which will force us to collect more data or use some data augmentation methods. Besides, although the research focus of this paper is the detection of license plates in unconstrained scenarios, the purpose of detection is for subsequent character recognition. Therefore, an end-to-end model that includes both detection and recognition is worth further exploration. In addition, due to the different character arrangement rules, the classification of license plates in different regions should also be taken into account. Finally, the license plate detection task under hard conditions such as low resolution, poor lighting, and accidental occlusion is still a major challenge to be solved.

CRediT authorship contribution statement

Chao Wei: Writing – review & editing, Writing – original draft, Conceptualization. **Fei Han:** Writing – review & editing, Supervision, Software. **Zizhu Fan:** Writing – review & editing, Writing – original draft, Supervision, Software, Resources, Project administration, Conceptualization. **Linrui Shi:** Visualization, Supervision, Funding acquisition, Formal analysis, Conceptualization. **Cheng Peng:** Investigation, Formal analysis, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: ZiZhu Fan reports financial support was provided by National Natural Science Foundation of China.

Data availability

The authors do not have permission to share data.

References

[1] L. Xie, T. Ahmad, L. Jin, Y. Liu, S. Zhang, A new CNN-based method for multi-directional car license plate detection, IEEE Trans. Intell. Transp. Syst. 19 (2) (2018) 507–517.

[2] C. Liu, F. Chang, Hybrid cascade structure for license plate detection in large visual surveillance scenes, IEEE Trans. Intell. Transp. Syst. 20 (6) (2019) 2122–2135.

[3] M. Molinamoreno, I. Gonzalezdiaz, F. Diazdemaria, Efficient scale-adaptive license plate detection system, IEEE Trans. Intell. Transp. Syst. 20 (6) (2019) 2109–2121.

[4] H. Caner, H.S. Gecim, A.Z. Alkar, Efficient embedded neural-network-based license plate recognition system, IEEE Trans. Veh. Technol. 57 (5) (2008) 2675–2683, http://dx.doi.org/10.1109/TVT.2008.915524.

[5] C.E. Anagnostopoulos, License plate recognition: A brief tutorial, IEEE Intell. Transp. Syst. Mag. 6 (1) (2014) 59–67, http://dx.doi.org/10.1109/MITS.2013.2292652.

[6] C.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos, E. Kayafas, License plate recognition from still images and video sequences: A survey, IEEE Trans. Intell. Transp. Syst. 9 (3) (2008) 377–391, http://dx.doi.org/10.1109/TITS.2008.922938.

[7] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K.M. von Deneen, P. Shi, An algorithm for license plate recognition applied to intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 12 (3) (2011) 830–845, http://dx.doi.org/10.1109/TITS.2011.2114346.

[8] A.H. Ashtari, M.J. Nordin, M. Fathy, An Iranian license plate recognition system based on color features, IEEE Trans. Intell. Transp. Syst. 15 (4) (2014) 1690–1705, http://dx.doi.org/10.1109/TITS.2014.2304515.

[9] M. Nejati, A. Majidi, M. Jalalat, License plate recognition based on edge histogram analysis and classifier ensemble, in: 2015 Signal Processing and Intelligent Systems Conference, SPIS, Tehran, Iran, 2015, pp. 48–52, http://dx.doi.org/10.1109/SPIS.2015.7422310.

[10] Xiuxia Yu, Hongyu Cao, Haidong Lu, Algorithm of license plate localization based on texture analysis, in: Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering, TMEE, Changchun, China, 2011, pp. 259–262, http://dx.doi.org/10.1109/TMEE.2011.6199192.

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

[12] Y. Chen, C. Zhao, T. Sun, Single image based metric learning via overlapping blocks model for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[13] Z. Cai, N. Vasconcelos, Cascade R-CNN: Delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 6154–6162.

[14] W. Wang, J. Yang, M. Chen, P. Wang, A light CNN for end-to-end car license plates detection and recognition, IEEE Access 7 (2019) 173875–173883, http://dx.doi.org/10.1109/ACCESS.2019.2956357.

[15] H. Li, P. Wang, M. You, C. Shen, Reading car license plates using deep neural networks, Image Vis. Comput. 72 (2018) 14–23, http://dx.doi.org/10.1016/j.imavis.2018.02.002.

[16] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, L. Huang, Towards end-to-end license plate detection and recognition: A large dataset and baseline, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 255–271.

[17] S.M. Silva, C.R. Jung, License plate detection and recognition in unconstrained scenarios, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 580–596.

[18] S.M. Silva, C.R. Jung, A flexible approach for automatic license plate recognition in unconstrained scenarios, IEEE Trans. Intell. Transp. Syst. (2021).

[19] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, CenterNet: Keypoint triplets for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2019, pp. 6569–6578.
[20] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.
[21] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, in: Proceedings of the European Conference on Computer Vision, ECCV, 2016, pp. 21–37.
[23] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.
[24] J. Redmon, S.K. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 779–788.
[25] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Jul. 2017, pp. 7263–7271.
[26] S. Montazzolli, C. Jung, Real-time Brazilian license plate detection and recognition using deep convolutional neural networks, in: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI, Niteroi, Brazil, 2017, pp. 52–62.
[27] H. Law, J. Deng, CornerNet: Detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 734–750.
[28] S.M. Silva, C.R. Jung, Real-time license plate detection and recognition using deep convolutional neural networks, J. Vis. Commun. Image Represent. 71 (2020) 102773.
[29] R. Laroca, E. Severo, L.A. Zanlorensi, L.S. Oliveira, G.R. Gonçalves, W.R. Schwartz, D. Menotti, A robust real-time automatic license plate recognition based on the YOLO detector, in: 2018 International Joint Conference on Neural Networks, IJCNN, 2018, pp. 1–10.
[30] R. Laroca, L.A. Zanlorensi, G.R. Gonçalves, E. Todt, W.R. Schwartz, D. Menotti, An efficient and layout-independent automatic license plate recognition system based on the YOLO detector, IET Intell. Transp. Syst. 15 (4) (2021) 483–503.
[31] H. Li, C. Shen, Reading car license plates using deep convolutional neural networks and LSTMs, 2016, arXiv preprint arXiv:1601.05610.
[32] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, P. Zemčík, Holistic recognition of low quality license plates by CNN using track annotated data, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS, IEEE, 2017, pp. 1–6.
[33] G. Hsu, J. Chen, Y. Chung, Application-oriented license plate recognition, IEEE Trans. Veh. Technol. 62 (2) (2013) 552–561, http://dx.doi.org/10.1109/TVT.2012.2226218.
[34] H. Li, P. Wang, C. Shen, Toward end-to-end car license plate detection and recognition with deep neural networks, IEEE Trans. Intell. Transp. Syst. 20 (3) (2018) 1126–1136.
[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[36] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Proceedings of the European Conference on Computer Vision, ECCV, 2014, pp. 740–755.
[38] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2961–2969.
[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988.
[40] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, E. Magli, Robust license plate recognition using neural networks trained on synthetic images, Pattern Recognit. 93 (2019) 134–146.
[41] Z. Selmi, M.B. Halima, U. Pal, M.A. Alimi, DELP-DAR system for license plate detection and recognition, Pattern Recognit. Lett. 129 (2020) 213–223.
[42] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, 2013, pp. 554–561.
[43] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[44] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017, arXiv preprint arXiv:1711.05101.