IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 July 2023, accepted 29 July 2023, date of publication 2 August 2023, date of current version 16 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3301122

RESEARCH ARTICLE

An Efficient License Plate Detection Approach With Deep Convolutional Neural Networks in Unconstrained Scenarios

WEI JIA AND MINGSHAN XIE, (Member, IEEE)
College of Big Data and Information Engineering, Guizhou University (GZU), Guiyang 550025, China

Corresponding author: Mingshan Xie (hnmingshanxie@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62266010, in part by the Cultivation Project of Guizhou University under Grant [2019]57, in part by the Research Project of Guizhou University for Talent Introduction under Grant [2019]31, in part by the Higher Education Research Project of Guizhou University under Grant GDGJYJ2020014, in part by the First-Class Curriculum Cultivation Project of Guizhou University under Grant XJG2021023, and in part by the Laboratory Opening Project of Guizhou University under Grant SYSKF2023-021.

ABSTRACT License plate (LP) detection is a crucial task for Automatic License Plate Recognition (ALPR) systems. Most existing LP detection networks can detect License plates, but their accuracy suffers when license plates (LPs) are tilted or deformed due to perspective distortion. This is because these detectors can only detect the region where the LP is located, and even the most advanced object detectors struggle in unconstrained scenarios. To address this problem, we propose a lightweight Deformation Planar Object Detection Network (DPOD-NET), which can correct the deformed LPs of various vehicles (e.g., car, truck, electric motorcycle, bus) by detecting the LP corner points. Accordingly, the distortion associated with perspective is mitigated when we adjust the LP to a frontal parallel view through the LP corners. To optimize small errors between the predicted and true values of the LP corner points, we propose an LPWing loss function. Compared with the commonly used L1 function, the LPWing loss is derivable at the zero position, and the gradient will be bigger when errors are smaller. This enables the model to converge faster at the position where the error is close to zero, resulting in better convergence when the error between the true values and predicted values is small. In addition, the paper presents a stochastic multi-scale image detail boosting strategy, which effectively augments the dataset. Finally, to objectively evaluate the effectiveness of LP corner detection approaches, we present a dataset (LPDE-4K) including various LP types (e.g., color, country, illumination, distortion). We test the performance on various datasets, and our approach outperforms other existing state-of-the-art approaches in terms of higher accuracy and lower computational cost.

INDEX TERMS License plate (LP) detection, convolutional neural networks, unconstrained scenarios.

I. INTRODUCTION

In recent years, Automatic License Plate Recognition (ALPR) technology has made great strides and has been widely adopted in various scenarios, including underground and open-air car parks. To improve the accuracy of LP recognition, many studies have focused on high-precision LP detection. Several LP detection approaches are available, which have shown high accuracy in controlled environments [1], [2], [3]. However, LP detection in unconstrained scenarios remains a challenge due to the diversity of environmental conditions.

There are two primary LP detection architectures: two-stage networks (LP detection and LP correction) and single-stage networks [2], [3], [4].The focus of these LP detection frameworks is to ensure complete LP information from a given scene and to correct the extracted LP information.

An efficient LP detection framework should be capable of detecting LPs in unconstrained scenarios. In this paper, we propose a two-part LP detection framework consisting of an efficient YOLOv5 vehicle detection network and our

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

proposed a lightweight Deformation Planar Object Detection Network (DPOD-NET).YOLOv5 is used to detect vehicles in unconstrained scenarios, which narrows the range of LP detection, while DPOD-NET is used for LP detection and correction. Therefore, our research focuses on designing a lightweight Deformation Planar Object Detection network that can detect vehicle LPs. Furthermore, to address the problem of making the prediction of the LP corner point closer to the actual value, we use a novel LPWing loss function to improve the network training process. This function is particularly useful and helps to optimize the error between predicted corner points and actual values. To augment the dataset, we propose a stochastic multi-scale image detail boosting strategy. As an additional contribution, we introduce an LP corner detection evaluation dataset and the LP corner points evaluation approach (LP-NME).

In summary, our contributions are as follows:

(1)A lightweight Deformation Planar Object Detection Network (DPOD-NET)is capable of correcting various distorted LPs by detecting LP corner points. The distortion associated with perspective is mitigated as we adjust the license plate to a frontal-parallel view through the LP corners.

(2)A novel Wing loss function for LP detection, comparing to the L1 function, the LPWing loss function is differentiable at zero and shows improved convergence when the error between the true values and predicted values is small. Experimental results demonstrate that this method surpasses others in terms of model accuracy and other aspects, all while maintaining network efficiency.

(3)Data Augmentation, a stochastic multi-scale image detail boosting approach is proposed. Experimental results demonstrate the advantages of the approach in improving image detail sharpness and image quality, and our approach effectively augmented the dataset.

(4)A LP detection evaluation dataset, LPDE-4K, containing images captured from a wide range of unconstrained scenarios. As an additional contribution, we try to objectify the LP corner points for evaluation and propose an evaluation approach for LP corner detection (LP-NME), which is described in Sections IV and V.

The remaining sections of the paper are organized as follows. Section II provides an overview of the work related to LP detection. Section III describes the approach proposed in this paper. Section IV explains the LPDE-4K dataset we have collected, and Section V presents a series of experimental results. Finally, we provide a systematic summary and future work in Section VI.

II. RELATED WORK

A. LP DETECTION

In the past few years, significant progress has been achieved in the field of computer vision, particularly in object detection and recognition [5], [6], [7]. However, in the area of license plate (LP) detection, most of the proposed approaches are still based on traditional image detection methods, such as region-based approaches, pixel-to-pixel approaches [8], and color-based approaches [9].These techniques have undoubtedly advanced the LP detection technology, but their accuracy in unconstrained scenarios is limited. Recently, more researchers have started to utilize convolutional neural network (CNN) to solve the LP detection problem [2], [10]. Several novel approaches have been proposed, such as Hsu et al.'s [11] LP detection algorithm based on YOLO [12] and YOLOv2 [13], their approach demonstrated good performance in precision and recall. However, the lack of objective evaluation of bounding boxes and the poor performance for small object detection in unconstrained scenarios need further investigation. Li and Shen [14] used deep neural network (DNN) in a cascade framework to extract LPs. Later, they proposed an end-to-end network for the final task [15]. Björklund et al. [16] presented a CNN model trained solely on synthetic LP images. They trained an LP corner detector for a regression model on three different datasets of LP images from various countries and transformed them into frontal-parallel views. Although their experimental results were excellent, retraining was required for different types of LPs. Selmi et al. [2] utilized image preprocessing technologies, such as image morphological processing and edge detection, to find candidate LPs in scenarios. Then, two different CNN networks were utilized for LP detection and recognition, respectively. However, this approach is sensitive to distorted LP and ambient illumination. Xie et al. [17] proposed two YOLO-Based detection networks for rotating LPs, the first network was used to locate the region of the LP, and the other network captured a rotating rectangular box. In their work, the author only considered the plane rotation, and experimental result was not the original shape of the LP. Laroca et al. [18] leveraged YOLO to detect LPs and constructed an ALPR system, which achieved superior performance on SSIG dataset.

With the recent advances in object detection algorithms, there has been a growing interest in using these models for license plate (LP) detection. However, the conventional detection approach, which only returns a rectangular box including two points, is not suitable for LP correction, and can negatively impact the robustness of LP detection. To address this issue, LP detection based on LP corner points has become increasingly popular, as this approach can accurately rectify deformed LP. Silva and Jung [19] proposed a new convolutional neural network (WPOD-NET) that can correct the distorted plane of LP by extracting LP corner points. WPOD-NET has achieved better performance on datasets such as AOLP. To further improve the performance of WPOD-NET, Silva et al. [20] proposed an improved LP correction network (IWPOD-NET), which achieves better performance on AOLP by adding network parameters. However, as the model becomes larger, the running speed becomes slower than WPOD-NET. Additionally, the authors used YOLOv3 [21] as a vehicle detector, which limited the LP detection performance. Therefore, the use of LP corner points for LP correction can improve the performance of automatic license plate recognition (ALPR) systems. In fact, similar approaches

have been widely studied in the field of face recognition. For example, face detection is performed first, and then face alignment is performed through face key points [22], [23]. Qi et al. [24] proposed a face detection approach based on YOLOv5, where they used the Wing loss function [25] to perform key point regression. It should be noted that the algorithm can also be used for the LP corner point detection. However, there is currently no evaluation approach for LP corner point detection. In this paper, we propose the LP-NME evaluation approach to address this issue. The experimental results demonstrate the effectiveness of the LP-NME evaluation approach.

B. LP DETECTION DATASETS

There are numerous datasets available for training and testing LP detection models, including well-known ones such as AOLP [11], SSIG [27], Open ALPR (which can be found at https://github.com/openalpr/benchmarks), CD-HA RD [19], UFPR [18], and CCPD [28]. However, it's worth noting that among these datasets, only CCPD features LPs marked with four corners, but it only includes Chinese LPs and with just one color. Unfortunately, a comprehensive dataset for evaluating LP critical corner points does not currently exist.

To tackle this issue, we have collected the LPDE-4K dataset, a medium-sized LP corner points dataset. Our dataset comprises images from various sources, including on-the-spot acquisitions, and contains challenging images captured in unconstrained scenarios, such as those with motion blur, uneven lighting, large bevel angles, and low resolution. We also included images of electric motorcycles, trucks, and buses in unconstrained scenarios. LP colors in our dataset aren't limited to blue and include other colors such as white and yellow. Additionally, the LP types in our dataset include Chinese, Taiwanese, British, Brazilian, European, and American LPs.

III. THE PROPOSED APPROACH

The entire pipeline of our License plate (LP) detection is shown in Fig. 1. This pipeline mainly consists of two modules:

(1) Vehicle Detection.

(2) Deformation Planar Object Detection Network (DPOD-NET).

In the following of this section, we will introduce in detail vehicle detection approach and the lightweight Deformation Planar Object Detection Network (DPOD-NET).

A. VEHICLE DETECTION

There are numerous outstanding object detectors available, considering factors such as handling diverse capturing scenarios, vehicle types, and operational efficiency, we have selected yolov5s6 as the vehicle detector for this paper. To accommodate the possibility of significant variations in image dimensions generated by the yolov5s6 model, we resize the input image dimensions prior to executing DPOD-NET, aiming to include as much of the license plate

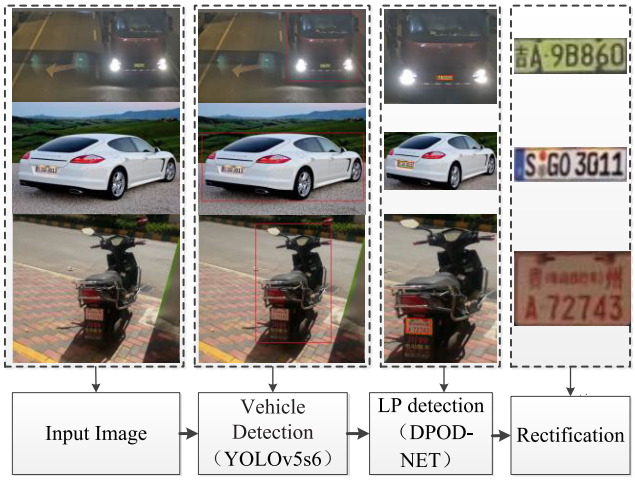

FIGURE 1. Illustration of the proposed pipeline.

image as possible. In most frontal or rear views, the width of the license plate experiences relatively limited changes compared to the cropped width of the vehicle. As a result, the vehicle bounding box tends to have an aspect ratio close to 1. However, in oblique views, the aspect ratio of the vehicle bounding box tends to be smaller due to the inclusion of the vehicle's side view in the image.

Based on the above analysis and related datasets(they will be discussed in Section V-A), We used scaling factor [20] F_{sc} to adjust the license plate image, and F_{sc}can be given by

$$F_{sc} = \left\{1, \frac{W_a}{w_a}\max\left\{1, \frac{w_a}{h_a}\right\}\right\} \tag{1}$$

where $w_a \times h_a$ are the bounding box dimensions of a detected vehicle, and W_a is a scaling constant (set to 256 for cars,buses, trucks and 208 for motorcycles).

B. DEFORMATION PLANAR OBJECT DETECTION NETWORK (DPOD-NET)

The process of utilizing DPOD-NET is depicted in Fig. 2. DPOD-NET takes the images produced by the vehicle detector as input and generates a 7-channel feature map (r_1-r_7). Among these channels, r_1 represents the probability map for distinguishing between license plates and non-license plates, while r_2-r_7 contain the feature maps for affine transformation parameters. To extract distorted license plates (LP), we define a fixed-sized virtual square with the center cell coordinates (m, n). If the object probability of that particular cell surpasses the designated detection threshold, we construct an affine matrix using the regression parameters and convert the square into the LP region.

(1).Network Architecture

Fig.3. illustrates the complete architecture of our network, which primarily comprises modular CBH convolution layers, CB convolution layers, and DPOD block layers. The convolution filters in our network have sizes of 3×3 and 1×1, and their number ranges from 16 to 128. We use

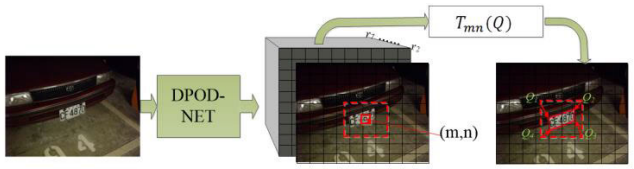

FIGURE 2. License plate detection and rectification process.

the Hard-Swish activation function [29] throughout the network. The CB convolution layer stride is set to 2 across the entire network. The last layer of our network consists of two branches: one is designed for LP location, returning the LP probability, while the other is responsible for extracting six affine transformation parameters, which are activated using the sigmoid function.

(2).DPOD Block

In recent years, the development of lightweight networks such as ShuffleNet v1 has garnered significant attention [29], [30], [31]. ShuffleNet v1 addresses issues related to information flow and weak representation between channel groups. In the DPOD Block of our network, we first perform a channel shuffling operation on the input features followed by a layering operation. We use a computationally economical 3×3 universal convolution module and a 1×1 convolution module [32] for one of the branch feature maps and then perform information stacking. Finally, we concatenate the information of both branches and perform another channel shuffling operation. The DPOD Block is illustrated in Figure 3(b).

(3).Transformer Encoder Block

Based on the vision transformer model [33], [34], we have incorporated a transformer encoder block into our network to enhance its performance. The block is illustrated in Fig.3(c) and is capable of extracting both global and contextual information. Each transformer encoder consists of two sub-layers: a multi-attention layer and a fully connected layer (MLP). Residual connections are applied between each sub-layer to improve the network's overall performance. By utilizing the transformer encoder block, our model can effectively capture diverse local information, while also exploring the potential of self-attentive mechanisms [35].

C. THE LOSS FUNCTION

(1).The affine transform and normalization function

As described by Silva and Jung [19] the definition of the LP corner points can be expressed as follows. Starting from the top left LP corner points, define the annotated LP corner points as $P_i = [x_i, y_i]^T$, for $i = 1, \ldots, 4$, correspondingly, define $Q_1 = [-0.5, -0.5]^T$, $Q_2 = [0.5, -0.5]^T$, $Q_3 = [0.5, 0.5]^T$, $Q_4 = [-0.5, 0.5]^T$ as the corner points of the unit square centered at the origin. Assuming that the RGB image is represented as $I \in R^{3 \times H \times W}$ and the network step $N_s = 16$, the network output feature map consists of $M \times N \times 8$ volumes, then $M = H/N_s$ and $N = W/N_s$.corresponding to each point cell (m, n) in the feature map, the network outputs seven estimates: the first (r_1) represents the probability of object/non-object, and the remaining values (r_2 *to* r_7) are used to parameter values of the affine transform for LP correction, so the affine transform T_{mn} can be defined as follows.

$$T_{mn}(Q) = \begin{bmatrix} r_2(n,m) & r_3(n,m) \\ r_4(n,m) & r_5(n,m) \end{bmatrix} Q + \begin{bmatrix} r_6(n,m) \\ r_7(n,m) \end{bmatrix} \quad (2)$$

To match the resolution of the network output, the LP corner points P_i are re-scaled by the inverse of N_s and shifted by (m, n) units to re-center the feature map points. This is accomplished by applying a normalization function.

$$A_{mn} = \frac{1}{\alpha}\left(\frac{1}{N_s}P - \begin{bmatrix} n \\ m \end{bmatrix}\right) \quad (3)$$

where α is a scaling constant that represents the side of the fictional square. We set $\alpha = 7.75$, which is the average point between the maximum and minimum LP dimensions in the augmented training data divided by N_s.

(2). LPWing loss function

Key point regression tasks commonly use popular loss functions such as L2, L1, and smooth-L1 [7]. For instance, the widely-used MTCNN [22] model adopts the L2 loss function for face key points regression. However, these loss functions tend to be less sensitive to small errors. This drawback has led to the development of new loss functions that can improve the network's training ability in small-medium ranges near zero. Inspired by recent research articles [22], [23], [24], [25], [36], we propose the LPWing loss function as a solution. The LPWing loss function is defined as follows

$$\text{LPWing}(x_k) = \begin{cases} Kx_k^2/2\mu |x_k| \le \mu \\ |x_k| - C |x_k| \ge \omega \\ \omega * ln(1 + |x_k|/\epsilon) - A\mu < |x_k| < \omega \end{cases} \quad (4)$$

where x_k is the error between corner points of the warped plane and the annotated LP corner points, the positive parameter μ is set to range of $[-\mu, \mu]$ and is primarily used for training very small values. For the small value of x ($x \in [\pm\mu, \pm\omega]$), we use the ln function and limit the curvature of the nonlinear region by parameter ε, $K = \omega / (\varepsilon + \mu)$, $A = \omega * ln(1 + \mu/\varepsilon) - K * \mu/2$and $C = \omega + \omega * ln((\varepsilon + \mu)/(\varepsilon + \omega)) - \omega * \mu/2 * (\varepsilon - \mu)$are used to adjust the linear and nonlinear smoothing constants. It is important to note that the value of ω should not be set too large or too small as this can lead to network instability and an explosion problem. In addition, we use x_k^2 in the zero-entry part to ensure the continuity of the gradient. Fig.4 shows the LPWing loss function under different parameter settings and the curves of different loss functions. The LPWing loss function behaves similarly to the R-Wing loss [36], but it is not set to zero in the interval $[-\mu, \mu]$. Fig.5(a) shows different loss functions, and Fig.5(b) shows their corresponding gradient curves. For L1, the gradient magnitude is consistent across all points, but the step size is disproportionately affected by large errors. On the other hand, L2 maintains a uniform step size while

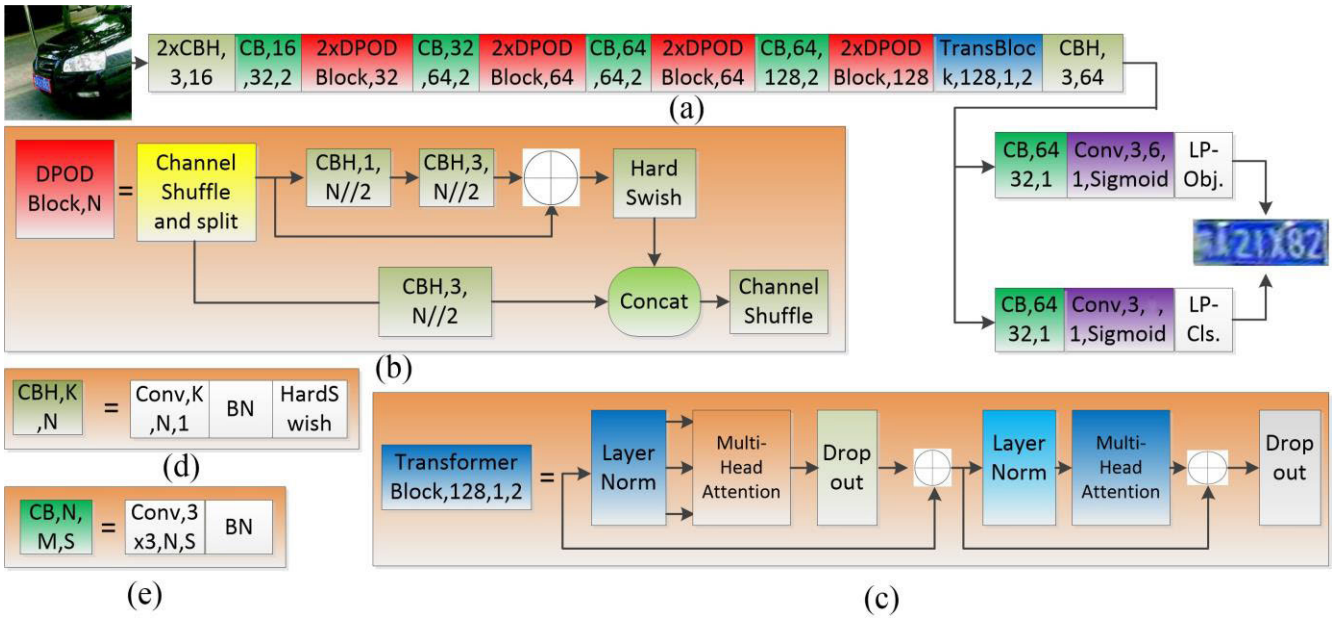

FIGURE 3. Overview of DPOD-NET, The input image size is 208 × 208. (a) DPOD-NET: This is the overall architecture of DPOD-NET. (b) DPOD Block: This is a specific block used in DPOD-NET. The letter "N" represents the number of channels. (c) Transformer Encoder Architecture: This component is comprised of two main blocks - a multi-head attention block and a feed-forward neural network (MLP). (d) CBH Convolution Layer: This layer uses a 3 × 3 filter with a stride of 1. (e) CB Convolution Layer: This layer uses a 3 × 3 filter with a stride of 2.

the gradient is influenced by large errors. Smooth L1 is a hybrid of L1 and L2, but it struggles to effectively correct relatively small errors. To account for the impact of minor errors, the *lnx*function can be employed, with a gradient of $1/x$. However, when x becomes extremely small, the gradient transformation becomes significant. Hence, when $|x_k| \leq \mu$, we resort to using L2 loss. Fig.5(b) shows that the gradient of the LPWing loss function is smooth and continuous, which is beneficial for training data that is close to zero.

In our experiment, we set the value of μ =0.1 and discovered that the model achieved its highest level of effectiveness when ω was set to 6 and ε was set to 2. For a more in-depth analysis of the impact of different parameter settings, please refer to Section V, where we present detailed ablation studies.

Fig. 2. illustrates the mapping relationship (indicated by red arrows) between the four points of the square and the four corner points of the license plate. We use the *LPWing* loss function to calculate the error between the four points of the square and the four corner points of the license plate. Let $x_k = \mathrm{T}_{mn}(Q_k P_)$ - $\mathrm{A}_{mn}(P_k)$, then the total loss for the four points can be expressed as

$$F_{loc}(m, n) = \sum_{k=1}^{4} LPWing(x_k) \tag{5}$$

(3). LP classification loss function

To predict whether there is LP at (m, n), we use focal loss [37], which can be expressed as follows:

$$F_{clas}(y, P) = -\alpha\ (1 - P)^{\gamma}\, y log(P) - (1 - \alpha) P^{\gamma}\ (1 - y) log(1 - P) \tag{6}$$

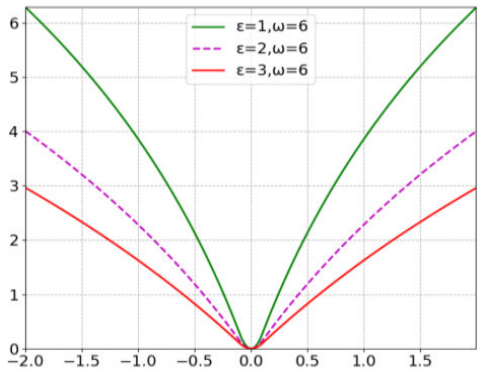

FIGURE 4. Curves of the LPWing loss functions, where ω =6.

where γ is called focusing parameter, $\gamma = 2$, y∈[+1, −1] for ground truth category, $P \in [1, 0]$ for the probability of LP, $\alpha \in [1, 0]$ for weight factor, α =0.25.

The final loss function is given by a combination of the terms defined in (5) and (6):

$$Loss = \sum_{m=1}^{M} \sum_{n=1}^{N} [wc * F_{clas}(m, n) * wl * F_{loc}(m, n)] \tag{7}$$

where wc =0.5, wl=0.5.

D. DATA AUGMENTATION AND TRAINING

To make the trained model better adapt to various complex environments, we employ several data augmentation techniques. Firstly, we introduce geometric changes such as adjusting the size and perspective of the images. Next, we use

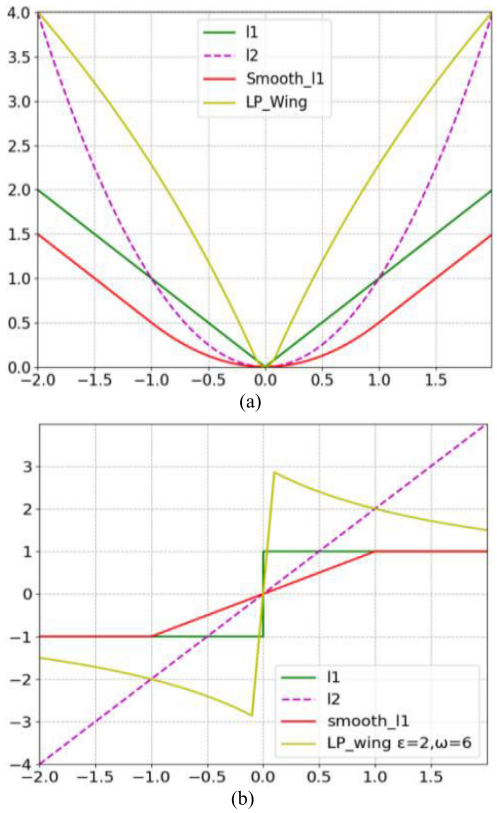

FIGURE 5. (a) Curve of the L2, L1, smooth L1 and LPWing loss functions. (b) Curve of the derivatives of different loss functions. We set the parameters $\omega = 6$, $\varepsilon = 2$.

color transformations by randomly modifying the HSV color space with a probability of p=100%. Additionally, we apply a Gaussian fuzzy approach with a probability of p=15%, where the degree of fuzziness σ is selected randomly within the range of [0, 0.1N]. In addition, we propose a stochastic image multi-scale detail boosting approach that is used to boost the training set image detail and augment the dataset. The next description elaborates this process.

In this work, the training set comprises diverse types of vehicles and license plates (LPs) sourced from different countries, which results in varying shapes of LPs. To tackle this challenge, we adopt the approach proposed by Silva et al. [20]. However, for LPs with large differences in length and width, we define the aspect ratio as $a_r \in [2.5, 5]$, and for LPs with close length and width (e.g., electric motorcycles), we define the aspect ratio as $a_r \in [1.25, 2.5]$ (e.g., electric motorcycles). Then, we choose angles of ± 55, ± 75, and ± 75 to produce LP samples with various 3D rotations. Finally, an augmented image is formed by distorting the input training image and gaze points, as well as adding a random background.

To further improve the quality of the transformed data, we have improved upon the multi-scale image detail boosting approach suggested by Kim et al. [38]. Specifically, we utilize Gaussian kernels to create four variations of the globally sourced image I, each with a different level of blur. This process enables us to obtain four differently blurred images that can be used to boost the detail of the original image at various scales.

$$B_1 = G_1 + I, B_2 = G_2 + I, B_3 = G_3 + I, B_4 = G_4 + I$$

where G_1, G_2, G_3, and G_4 are the Gaussian kernels with the standard deviations $\sigma_1 = 1.0$, $\sigma_2 = 2.0$, $\sigma_3 = 4.0$, $\sigma_4 = 6.0$ respectively. We extract all of the details by

$$\begin{aligned} D_1 = I - B_1, D_2 = B_1 - B_2, D_3 = B_2 - B_3, D_4 \\ = B_1 - B_3, D_5 = B_1 - B_4, D_6 \\ = B_2 - B_4, D_7 = B_3 - B_4 \end{aligned}$$

Then, the overall detail image can be defined by

$$\begin{aligned} D^* = (1 - \omega_1 tanh(D_1)) * D_1 + \omega_2 D_2 + \omega_3 D_3 + \omega_4 D_4 \\ + \omega_5 D_5 + \omega_6 D_6 + \omega_7 D_7 \end{aligned} \quad (8)$$

where ω_1, ω_2, ω_3, ω_4, ω_5, $\omega_6$and$\omega_7$are fixed to 0.5, 0.5, 0.25, 0.5, 0.5, 0.5, 0.25, 0.125 respectively. Note that the image may appear oversaturated when simple superposition is applied. Therefore, we utilize the hyperbolic tangent function approach to balance the image pixels [39], which helps to suppress image oversaturation. In Fig.6, it can be observed that our approach outperforms the original approach.

Our data augmentation approach involves specifying the Gaussian kernel size (ksize) as a random value, enabling the generation of different detail-boosted images. This process increases the amount of data in the original training set to some extent. In Fig.7, we present the results of our experiments for selecting various Gaussian kernel sizes. Notably, when ksize=1, the detail-boosted image is not evident. However, as the ksize increases, the image detail becomes significantly better. During the model training phase, we set the value of ksize to a list [1, 1, 3, 5, 7, 9, 11], from which we randomly select a value as the size of the current Gaussian kernel.

With the use of data augmentation techniques described above, the variety and quality of the data are enhanced compared to the original dataset, as demonstrated in Fig.8 which shows the effect of the augmentation pipeline on a sample image.

To train our model, we created a small training set consisting of 842 images. The set comprises 125 images of electric motorcycles, 100 images of trucks, 124 images of buses, and cars with license plates of various colors. The input image size is 208×208. During the model training phase, we employed the adamax optimizer [40] with a batch size of 64 and an initial learning rate of 0.01. The configuration of the learning rate and iteration number in a training process

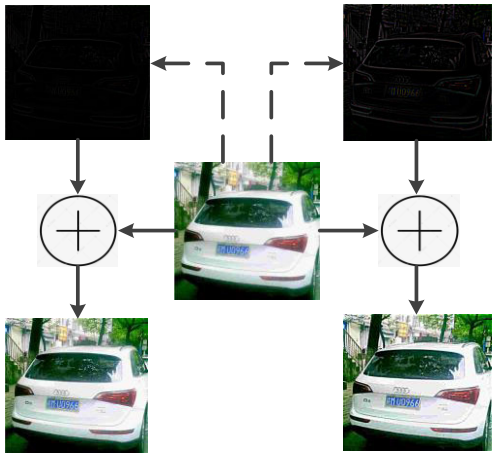

FIGURE 6. Comparison of the multi-scale image detail boosting effect, using a Gaussian kernel size of 5. The original image, obtained from CCPD, is shown in the center of the figure. The top-left corner of the figure displays the image detail extracted using the original approach, while the top-right corner shows the image detail extracted using our improved approach. The bottom-left corner of the figure shows the image generated by the original approach [38], while the bottom-right corner displays the image generated by our approach. Through this comparison, it is evident that our approach leads to a better boosting of image details, thereby improving the overall image quality.

can be expressed as

$$L_R = \begin{cases} 0.01 & N \leq 80k \\ \frac{0.01}{5} & N \leq 160k \\ \frac{0.01}{5^2} & N \leq 240k \\ \frac{0.01}{5^3} & N \leq 320k \\ \frac{0.01}{10^3} & N \leq 400K \\ \frac{0.01}{5 \times 10^3} & N \leq 560k \\ \frac{0.01}{10^4} & N \leq 880k \end{cases} \tag{9}$$

where L_R is learning rate, N is iteration number. Ultimately, we chose the model with the most optimal performance metrics, as outlined in Section V.

IV. LP CORNER POINTS DETECTION AND EVALUATION DATASET

The main datasets for LP detection are AOLP [11], SSIG [27], OpenALPR (available at https://github.com/openal

pr/benchmarks), CD-HARD [19], UFPR [18] and CCPD [28]. These datasets include few LP types. For example, CCPD includes a large number of car LPs, a small number of bus LPs, and only one LP color. AOLP includes only car LPs. However, as the number of electric vehicles increases worldwide, the detection of a large number of electric vehicle LPs becomes increasingly important. Therefore, we constructed a multi-vehicle dataset consisting of 4000 images. In the dataset, Vehicle types include electric motorcycle, car, truck and bus, LP colors consists of blue, red, yellow and white. The dataset also contains LPs from different countries, such as China, Brazil, Iran, USA, Taiwan, Europe, UK, etc. Besides, our dataset marks the four corners of LPs and thus can be used for LP detection and evaluation.

To construct the LPDE-4K dataset, we have taken into consideration various factors that affect the image quality, including illumination, angle, distance, and complex backgrounds. As a result, our proposed dataset is extremely challenging in all aspects. The majority of the images in LPDE-4K are sourced from unconstrained scenarios with varying image tilts, distances, illuminations, and blurs, making LP detection a difficult task. Furthermore, our dataset includes numerous real-world scenarios where LP detection may be required.

As shown in TABLE 1, our new LPDE-4K dataset encompasses a wide range of national LPs, vehicle types, and LP colors. Additionally, we have focused on coarse scenarios in uncontrolled environments, specifically including images with uneven lighting, angle changes, and low resolutions. Therefore, our dataset is particularly challenging.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The implementation of all our codes are based on the learning framework of PyTorch, we conducted our experiments on the NVIDIA GTX1070TI GPU and Intel Core I5-9400F CPU. Our proposed approach consists of two networks in the pipeline, and we have empirically established the acceptable thresholds for these networks. Specifically, we set the threshold for vehicle detection using YOLOv5s6 to 0.35, and the threshold for license plate (LP) detection using DPOD-NET to 0.45. It is important to note that our LP detection network is used to detect the region generated by the vehicle detector YOLOv5s6. After conducting multiple experimental tests, we chose a threshold of 0.3 for the vehicle detector to reduce the number of false negatives. All of our experimental results were generated using the default settings without any prior knowledge of the LP region. We also clearly described any post-processing methods used in our experiments.

A. DATASET

In this paper, our experiment aims to verify the effective-ness and robustness of an algorithm for LP detection. To evaluate the performance, we used four datasets: CCPD, AOLP, OPENALOP (available: https://github.com/ope

nalpr/bench-marks), and LPDE-4K.

(1)**AOLP [11]**is split into three categories: Tilt LP, Front LP, and General LP. These images are divided into three subsets: AC (681 samples), LE (757 samples), and RP (611 samples). In our experiments, we re-annotated the corner points of all LPs in the AOLP dataset.

(2)**CCPD [28]**is a large, diverse, and carefully annotated open-source dataset of Chinese city LPs. This dataset was

collected in the He Fei car park from 7:30 a.m. to 10:00 p.m. The LP images involve a variety of complex environments, including blur, tilt, rain, snow, and more. The dataset contains over 250,000 Chinese city LP images and

FIGURE 7. The experimental results of our approach in different Gaussian kernel sizes (ksizes), the first row is the original image, the second row is the extracted image details, and the last row is the detail-boosted image.

FIGURE 8. The final augmentation effect of the same image.

annotations for number plate detection and recognition information.

(3)**OpenALPR**(https://github.com/CarltonSemple/openalpr) is an open-source automatic LP recognition library that comes with a dataset containing over 800 LPs, including EU, BR, and US sub-datasets.

To train the YOLOv5s6 network, we collected a dataset of 60,000 vehicle images that include cars, buses, motorbikes,and trucks. These vehicle images were obtained from datasets such as CCDP [28], BDD100K (available: https://bdd-data.berkeley.edu), COCO2017 (available: https://cocodatataset.org/#home), VOC2012 (available: http://host.robots.ox.ac.uk/pasc-al/VOC/voc2012/), AOLP [11], SSIG [27], and the data we collected ourselves.

B. EVALUATION METRICS

To ensure the accuracy of our model, we utilized standard precision (PR), recall (RE), and F-Measure (F-M.) evaluation protocols to assess the performance of our model on the AOLP dataset. Moreover, we employed the same dataset to evaluate the effectiveness of the LPWing loss function. For evaluating the accuracy performance of our model, we utilized the CCPD benchmark.

Normalization Mean Error (NME) is commonly used to evaluate performance of face key points detection [26], [36]. As far as we know, there is no similar approach in LP corner point evaluation. Inspired by NME, we proposed an improved NME, namely LP-NME, and it is used as an evaluation approach in our experiment. The LP-NME for each LP image

TABLE 1. Comparison of key attributes in the license plate detection dataset:images: the total number of images included in the dataset.vehicle types: the color of the license plates (LPS) for different vehicle types.LP countries: LPS from various countries included in the dataset.Real scenes: instances of lps and non-lps present in real-world scenes.LP marking: the number of corner points that mark the LPS in the images.LPS/IMAGE: the average number of LPS present in each image.

Name(Year)	Images	Vehicle types	LP color	LP countries	LP marking(Corner point)	LPs/image
AOLP(2012)[11]	2049	1	1	1	2	1
SSIG(2016)[27]	2000	2	2	1	2	4.34
UFPR(2018)[18]	4500	4	4	1	2	1
CD-HARD(2018)[19]	102	1	1	1	✗	1
CCPD(2018)[28]	250K	2	1	1	2/4	1
QpenALPR	800	3	1	2	✗	1
LPDE-4K	4K	4	4	5	4	1

is defined as follows

$$LP-NME\left(P,\hat{P}\right)=\frac{1}{N}\sum_{i=0}^{N}\frac{\left\|P_i-\hat{P}_i\right\|}{d} \tag{10}$$

where $P$and $\hat{P}$ are the annotated LP corner points and the predicted LP corner points for each image, respectively. Additionally, N is the total number of LP corner points for each image, $d$is the normalization factor. For the AOLP dataset, we utilized the diagonal length of the smallest rectangular box that can enclose the LP as the normalization factor. An illustration of the smallest rectangular box is provided in Fig.9, and is represented in blue.

The LP-NME calculation is utilized to determine the average value of the relative error between the predicted and true corner points. It is crucial to note that selecting an appropriate normalization factor, denoted as d, is a critical aspect of this calculation. Typically, the d value in the NME formula is selected as the distance between the pupils of the human eye. In a similar manner, for LP-NME, the diagonal length of the smallest rectangular box that can surround the LP is used as the normalization factor. As the distance between the four predicted points relative to the annotated points increases, the LP-NME increases as well. Conversely, when the predicted value is closer to the true value, the LP-NME becomes smaller, indicating greater accuracy in predicting the corner points. TABLE 5 demonstrates that the model's performance is better when the RE, PR, and F-M. values are larger, and the LP-NME value is smaller.

C. LPWING LOSS FUNCTION PARAMETERS AND EVALUATION

To find the optimal parameter settings for LPWing loss function, we began by manually annotating the LP corner points of the AOLP dataset. Next, we explored different parameter combinations and evaluated their performance on the AOLP dataset. In order to narrow down the search space for the parameters, we set the initial values of ω and ε to 6 and 1, respectively, while the value of parameter μ was set to 0.1. We believed that setting the values of ω and ε too large could result in network instability, while setting them too small could cause the gradient explosion problem. We tested

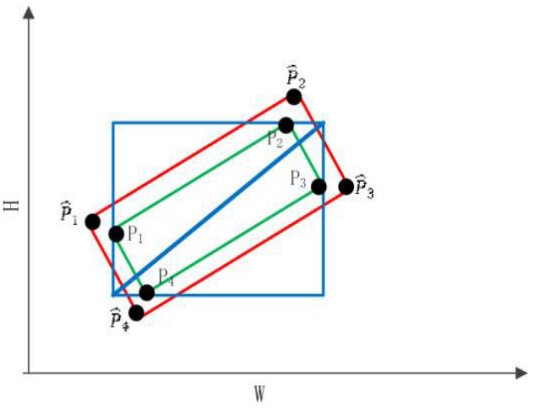

FIGURE 9. Illustration of the LP-NME used to evaluate the LP corner points. The LP corner points that are predicted are denoted by $\hat{P}_i$(i=1…4), while the annotated LP corner points are represented byP_i (i=1…4). The diagonal length of the blue rectangular box is denoted by d.

TABLE 2. Evaluation on different parameter ω settings of the lpwing loss.

ε / ω	6	8	10	12
1	0.88	0.91	0.91	0.94
2	**0.57**	1.11	0.89	0.90
3	0.88	0.79	1.63	0.72

different combinations of ω and ε, training DPOD-NET for 50k iterations each time, and evaluated the average LP-NME values on the AOLP dataset for each combination. The results of this experiment are presented in TABLE ?, which shows that our LPWing converges faster when ω =6 and ε =2.Once we had determined the optimal values for ω and ε, we fixed them and examined the performance of different loss functions (L1, L2, Smooth-L1 and LPWing). The results are presented in TABLE 3, which shows that the LPWing loss function resulted in the lowest LP-NME value, indicating that the predicted value was closest to the real value. TABLE 4 shows that LPWing consistently yields smaller LP-NME values compared to the L1 loss function across various models. Finally, we compared the performance of different loss functions for different network frameworks in TABLE 5.

FIGURE 10. comparison of our approach with WPOD-NET [19] and IWPOD-NET [20], the red box represents the detection result, while the green box indicates the actual box. The corrected LP and LP-NME values can be found in the lower left corner of the each image. The images used in this comparison are from the LPDE-4K and ALOP datasets.

TABLE 3. Based on the test results carried out by DPOD-NET, it has been observed that when the values of Ω and *E* are 6 and 2 respectively, the average LP-NME (AVG.) of various loss functions was calculated. The results indicate that lpwing loss function outperformed the other loss functions.

Approach	LP-NME
DPOD-NET+Smooth-L1[7]	1.39
DPOD-NET+L1	0.72
DPOD-NET+L2	1.00
DPOD-NET+LPWing	**0.57**

TABLE 4. LP-NME values of the L1 and LPwing loss functions across various models.

Approach \ Loss Function	L1	LPWing
IWPOD-NET [20]	0.95	**0.74**
DPOD-NET	0.72	**0.57**

D. COMPARISON OF LP DETECTION QUALITY WITH STATE-OF-THE-ART APPROACHES

In this section, we present the performance of our proposed approach compared to other state-of-the-art methods. To evaluate our approach, we employed precision (PE), recall (RE) rates, and F-Measure (F-M.), which are commonly used in LP detection. Since there is no objective evaluation standard for LP corner point detection, we also used LP-NME for performance evaluation, which has been widely used in LP detection in recent years. The experimental results demonstrate the reliability of LP-NME for LP corner point detection.

FIGURE 11. Partial samples of images, featuring a diverse range of objects. The first two images in the top row are sourced from the CCDP dataset, while the remaining images are sourced from the LPDE-4K dataset. The red box in each image represents the object detection result, while the green box indicates the ground truth bounding box. The corrected values of the LP and LP-NME metrics are displayed in the lower left corner of each image.

For the ALOP sub-dataset experiment, we did not use our trained YOLOv5s6 vehicle detector, but only compared the performance of our DPOD-NET. We re-annotated the ALOP dataset to include all LPs. In TABLE 5, the results show that our approach outperforms other advanced approaches. The precision of DPOD-NET is 99.9 on the AC subset, which is better than IWPOD. Additionally, the recall and F-M. exceed those of Björklund et al. [16]. On the LE and RP subsets, the average LP-NME has a smaller value while showing a higher precision, indicating that DPOD-NET performs better in terms of deformed planar object detection.

In our experiment, we selected five subsets from the CCPD dataset and considered the number of samples in each subset. TABLE 6 presents a comparison of detection accuracy results with existing LP detection approaches. The results indicate that DPOD-NET achieves the highest accuracy in LP corner point detection on the DB. subset, with a detection accuracy of up to 99.1 using the YOLOv5s6-DPOD-NET, surpassing other approaches. Although DPOD-NET also has the highest accuracy on the Rot., Tilt., and Weath. subsets, its accuracy on the Chall. subset is slightly lower than that of IWPOD-NET, with a detection accuracy of 92.7. However, compared to all the other approaches, YOLOv5s6-DPOD-NET outperforms them with the highest accuracy. Notably, the YOLOv5s6 vehicle detector reduces the complexity of the scenarios and produces more conducive results for LP detection with DPOD-NET. We trained the vehicle detector using 60,000 vehicle images from different datasets, whereas for DPOD-

TABLE 5. LP detection precision (IN %), recall (IN %), F-score (IN %) and LP-NME (AVG.)

Approach(Year) Subset	AC				LE				RP			
	PR	RE	F-M.	LP-NME	PR	RE	F-M.	LP-NME	PR	RE	F-M.	LP-NME
Hsu et. al.(2012)[11]	90.9	95.9	93.3	—	91.0	95.3	95.1	—	91.0	94.0	92.5	—
H. Li et al(2016)[14]	98.5	98.3	98.4	—	97.8	97.6	97.7	—	95.3	95.6	95.5	—
Björklund(2019)[16]	**100**	99.3	99.6	—	99.8	99.1	**99.5**	—	99.8	99.2	99.5	—
Xie L(2018)[17]	99.5	99.5	99.5	—	99.4	**99.4**	99.4	—	99.5	99.5	99.5	—
Selmi et.al.(2017)[2]	99.3	99.3	99.3	—	99.1	99.2	99.1	—	99.1	98.8	99.0	—
WPOD-NET(2018)[19]	94.0	99.1	96.5	0.76	99.6	93.3	96.5	0.96	99.8	96.8	98.3	0.45
IWPOD-NET(2021)[20]	99.7	99.7	99.7	0.12	99.5	96.2	97.8	0.57	99.8	99.8	99.8	0.14
DPOD-NET	99.9	**99.7**	**99.8**	**0.12**	**99.9**	97.4	98.6	**0.37**	**100**	**100**	**100**	**0.12**

TABLE 6. The LP detection accuracy (IN %) for CCPD.

Approach(Year) Subset	DB.(190k).	Rot.(10k).	Tilt.(30k).	Weath.(10k).	Chall.(50k).
Casocade (2007)[5]	49.2	42.1	60.1	51.5	27.5
SSD(2016)[6]	99.0	95.6	95.0	83.5	93.2
YOLO9000(2017)[13]	98.8	93.4	92.0	84.2	88.5
Faster-RCNN(2015)[7]	98.1	91.7	89.5	82.0	83.9
RPNet (2018)[28]	99.0	94.5	93.5	84.0	92.8
TE2E(2018)[15]	98.5	95.1	94.4	83.6	93.0
YOLOv3(2018)[21]	97.1	91.7	94.6	98.0	90.4
WPOD-NET(2018)[19]	97.5	94.3	92.6	96.9	89.9
IWPOD-NET(2021)[20]	98.6	95.2	93.2	95.9	93.5
DPOD-NET	98.8	**95.6**	**94.8**	97.6	92.7
YOLOv5s6-DPOD-NET	**99.1**	**96.5**	**97.9**	**98.8**	**96.0**

TABLE 7. LP detection precision (IN %), recall (IN %), F-score (IN %) and LP-NME (AVG.) on ALOP, openalop and LPDE-4K.

Approach(Year) Subset	ALOP				OPENALOP				LPDE-4K			
	AC	PR	RE	LP-NME	AC	PR	RE	LP-NME	AC	PR	RE	LP-NME
WPOD-NET(2018)[19]	96.8	99.5	94.6	0.750	94.0	94.6	93.4	0.910	81.2	94.2	79.4	2.32
IWPOD-NET(2021)[20]	98.6	99.7	98.4	0.295	96.7	99.0	94.2	0.673	95.0	97.6	89.0	1.02
DPOD-NET	**99.7**	**99.9**	**98.8**	**0.220**	**97.5**	98.2	**95.8**	**0.625**	**95.2**	95.3	**89.4**	**1.01**
YOLOv5s6-DPOD-NET	**99.9**	**100**	**99.8**	**0.180**	**99.0**	**99.2**	**96.8**	**0.580**	**98.5**	**99.7**	**94.3**	**0.92**

NET, we only used 842 images, including 420 from the CCPD dataset.

Furthermore, our approach's comprehensive detection performance on ALOP, OPENALOP, and LPDE-4K datasets is summarized in TABLE 7. Notably, our approach attains the best results on the ALOP dataset with LP-NME values of 0.220 and 0.180, respectively, signifying that the LP corner points are in closer proximity to the true values. On the OPENALOP and LPDE-4K datasets, DPOD-NET outperforms IWPOD-NET in all metrics except for precision. Finally, by incorporating the YOLOv5s6 vehicle detector into DPOD-NET, we are able to reduce LP-NME by 0.08 on the OPENALOP dataset and by 0.08 on LPDE-4K, and improve other metrics as well. Overall, these results demonstrate that our approach is most suitable for LP detection in unrestricted scenes.

TABLE 8. Complexity information of the models (input image size: 208) × 208.

Model(Year)	Model size (MB)	Parameters (MB)	Floating point Operations (FLOPs)	Detection speed (ms/Image)		
				GTX*1070Ti*	*I*5-10300H	I5-9400F
IWPOD-NET(2021)[20]	7.36	1.73	1.18G	9.6	232	223.1
DPOD-NET	**4.43**	**0.95**	**0.72G**	**9.3**	**85.5**	**81.1**
YOLOv3-IWPOD-NET(2021)[20]	250.00	42.30	25.70G	23.2	32696	25500
YOLOv5s6-DPOD-NET	28.90	13.60	17.50G	18.3	240	202

Fig.10. depicts electric motorcycles with white and yellow LP colors, which are almost square in shape. Our approach yields the best results with the smallest LP-NME value, whereas IWPOD-NET detects a false positive in the first row of electric motorcycle LP detection. In the last two rows of images, the LP color is white, and the LP is distorted. Our detection approach performs better and can locate the vehicle LP more accurately, as evidenced by the minimum LP-NME value.

To demonstrate the actual results of YOLOv5s6-DPOD-NET, Fig.11 presents some qualitative results. The first two images are from CCDP dataset, while the others are from LPDE-4K. We observe that YOLOv5s6-DPOD-NET works well in various scenarios, including in real road surveillance scenarios. Overall, our approach demonstrates superior performance in detecting vehicle LPs, particularly in challenging scenarios where the LP is distorted or has an unconventional shape. These results show the potential of our approach for real-world applications, such as traffic monitoring and law enforcement.

E. COMPUTATIONAL COMPLEXITY COMPARISON

TABLE 8 displays the complexity information for four models: IWPOD-NET [20], DPOD-NET, YOLOv3-IWPOD-NET [20], and YOLOv5s6-DPOD-NET. We discover that DPOD-NET has fewer parameters than IWPOD-NET, resulting in a 39% reduction in FLOPs. Moreover, the model size of DPOD-NET is a mere 4.43M.We also measured the average time required to run a single image on the ALOP dataset. Interestingly, our findings indicate that DPOD-NET's run time on I5-10300H (CPU) and I5-9400F (CPU) is 2.7 and 2.8 times faster than that of IWPOD-NET, respectively. It is worth noting that our proposed approaches demonstrate the shortest time on GPU and exhibit superior performance compared to other approaches.

VI. CONCLUSION AND FUTURE WORK

In this work, our focus was on studying the detection of deformed license plates (LPs) in unconstrained scenarios. To achieve this, we proposed an effective deformed planar object detection network (DPOD-NET), with several key contributions. Firstly, we designed a network module and subsequently proposed a plane distortion detection network, which can correct distorted LPs by detecting their corner points. Notably, our network was trained using only 842 images, yet experiments showed that it had good generalization ability across a variety of unconstrained scenarios. Moreover, our network demonstrated the capability to detect various types of LPs. Secondly, we introduced the LPWing loss function to enhance the training ability of the loss at positions close to the zero point during the network training process. This helped optimize the error between the predicted value and the actual value. To improve the quality of the dataset, we also proposed the image random multi-scale detail boosting algorithm used in data augmentation. Finally, we created a new benchmark for LP corner point evaluation called LPDE-4K.

Based on the experimental results, our proposed license plate (LP) detection approach demonstrates comparable or even better performance than state-of-the-art methods in all test metrics, even in challenging unconstrained scenarios. Notably, our approach outperforms all other methods when YOLOv5s6 is used as the vehicle detector. In addition to accurately detecting LPs, our approach can effectively handle plane distortion correction for electric motorcycles, cars, trucks, and buses. Surprisingly, our approach is also significantly faster than existing approaches, with a speed-up of 2.7 and 2.8 times on the CPU used in our experiments. Furthermore, our approach exhibits superior performance on the GPU, making it a highly efficient solution for LP detection.

In future work, we will be to design an efficient end-to-end license plate detection and correction network that can be applied to various embedded devices or vehicular edge computing devices [41], [42], such as open-air parking inspection robots, underground parking inspection robots, robot police, and vehicular edge computing.

REFERENCES

[1] J. Dun, S. Zhang, X. Ye, and Y. Zhang, "Chinese license plate localization in multi-lane with complex background based on concomitant colors," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 51–61, Fall. 2015.

[2] Z. Selmi, M. Ben Halima, and A. M. Alimi, "Deep learning system for automatic license plate detection and recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1132–1138. [Online]. Available: http://ieeexplore.ieee.org/document/8270118/

[3] Y. Y. Lee, Z. A. Halim, and M. N. A. Wahab, "License plate detection using convolutional neural network–back to the basic with design of experiments," *IEEE Access*, vol. 10, pp. 22577–22585, 2022.

[4] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102773.

[5] S.-Z. Wang and H.-J. Lee, "A cascade framework for a real-time statistical plate recognition system," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 2, pp. 267–282, Jun. 2007.
[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 21–37.
[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[8] J. Muhammad and H. Altun, "Improved license plate detection using HOG-based features and genetic algorithm," in *Proc. 24th Signal Process. Commun. Appl. Conf. (SIU)*, May 2016, pp. 1269–1272.
[9] Y. Wang, X. Ban, J. Chen, B. Hu, and X. Yang, "License plate recognition based on SIFT feature," *Optik*, vol. 126, no. 21, pp. 2895–2901, 2015.
[10] Z. Mahmood, K. Khan, U. Khan, S. H. Adil, S. S. A. Ali, and M. Shahzad, "Towards automatic license plate detection," *Sensors*, vol. 22, no. 3, p. 1245, 2022.
[11] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 552–561, Feb. 2013.
[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[14] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMs," 2016, *arXiv:1601.05610*.
[15] H. Li, P. Wang, and C. Shen, "Toward end-to-end car license plate detection and recognition with deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1126–1136, Mar. 2019.
[16] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Robust license plate recognition using neural networks trained on synthetic images," *Pattern Recognit.*, vol. 93, pp. 134–146, Sep. 2019.
[17] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 507–517, Feb. 2018.
[18] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Goncalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–10.
[19] S. Silva and C. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 580–596.
[20] S. M. Silva and C. R. Jung, "A flexible approach for automatic license plate recognition in unconstrained scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5693–5703, Jun. 2022.
[21] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[22] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[23] X. Guo, S. Li, J. Yu, J. Zhang, J. Ma, L. Ma, W. Liu, and H. Ling, "PFLD: A practical facial landmark detector," 2019, *arXiv:1902.10859*.
[24] D. Qi, W. Tan, Q. Yao, and J. Liu, "YOLO5Face: Why reinventing a face detector," 2021, *arXiv:2105.12931*.
[25] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2235–2245.
[26] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6971–6981.
[27] G. R. Gonçalves, S. P. G. D. Silva, D. Menotti, and W. R. Schwartz, "Benchmark for license plate character segmentation," *Proc. SPIE*, vol. 25, no. 5, Oct. 2016, Art. no. 053034.
[28] Z. Xu, W. Yang, A. Meng, N. Lu, and H. Huang, "Towards end-to-end license plate detection and recognition: A large dataset and baseline," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 255–271.
[29] B. Koonce, "MobileNetV3," in *Convolutional Neural Networks With Swift for TensorFlow*. Berkeley, CA, USA: Apress, 2021, pp. 125–144.
[30] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[31] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[32] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[34] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 1–11.
[36] Z.-H. Feng, J. Kittler, M. Awais, and X.-J. Wu, "Rectified wing loss for efficient and robust facial landmark localisation with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 128, nos. 8–9, pp. 2126–2145, Sep. 2020.
[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[38] Y. Kim, Y. J. Koh, C. Lee, S. Kim, and C.-S. Kim, "Dark image enhancement based onpairwise target contrast and multi-scale detail boosting," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 1404–1408.
[39] W. Jia, Y. Wang, Y. Liu, L. Fan, and Q. Ruan, "An adaptive enhancement algorithm of materials bag image of industrial scene," in *Proc. Conf. Intell. Robot. Appl. (CIAR)*, 2014, pp. 226–238.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*.
[41] C. Chen, G. Yao, C. Wang, S. Goudos, and S. Wan, "Enhancing the robustness of object detection via 6G vehicular edge computing," *Digit. Commun. Netw.*, vol. 8, no. 6, pp. 923–931, Dec. 2022.
[42] X. Deng, L. Wang, J. Gui, P. Jiang, X. Chen, F. Zeng, and S. Wan, "A review of 6G autonomous intelligent transportation systems: Mechanisms, applications and challenges," *J. Syst. Archit.*, vol. 142, Sep. 2023, Art. no. 102929.

WEI JIA received the master's degree in computer vision from Sichuan University, China, in 2013. He is currently a Lecturer with Guizhou University, China. His research interests include machine learning, intelligent systems, the Internet of Things robot, smart space, and artificial intelligence.

MINGSHAN XIE (Member, IEEE) received the B.S. degree in educational technology from Southwest University, Chongqing, China, in 2004, the M.S. degree in computer application technology from Hainan University, Haikou, China, in 2013, and the Ph.D. degree in information and communication engineering, in 2019. He is currently with College of Big data and Informaton Engineering, Guizhou University. His research interests include the Internet of Things robot, smart space, and artificial intelligence.