

AN EFFECTIVE METHOD FOR DETECTING LICENSE PLATES USING DEEP CONVOLUTIONAL NEURAL NETWORKS IN UNCONSTRAINED ENVIRONMENTS**Charitha Bandarupalli****Archana Bandi****Sravani Barma**B. Tech Students, Dept. of Computer Science and Engineering,
R.V.R & J.C College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India**Dr M.Srikanth**Professor, Dept. of Computer Science and Engineering,
R.V.R & J.C College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India

ABSTRACT

One of the most important tasks for Automatic License Plate Recognition (ALPR) systems is license plate (LP) detection. Although the majority of LP detection networks currently in use are capable of detecting license plates, perspective distortion causes these networks to lose accuracy when LPs are tilted or twisted. This is because even the most sophisticated item detectors have trouble in unrestricted situations, and these detectors are limited to detecting the area where the LP is located. In order to solve this issue, we suggest a lightweight Deformation Planar Object Detection Network (DPOD-NET), which uses LP corner point detection to rectify the distorted LPs of a variety of vehicles, including cars, trucks, electric motorcycles, and buses. Therefore, by adjusting the LP to a frontal parallel view through the LP corners, the perspective distortion is lessened.

We provide an LPWing loss function to minimize tiny discrepancies between the true and forecasted values of the LP corner points. The gradient will be larger when mistakes are less, thus the LPWing loss can be derived at the zero position in contrast to the widely used L1 function. This makes it possible for the model to converge more quickly at the point where the error is near zero, leading to improved convergence when there is little error between the true and predicted values. The study also offers a stochastic multi-scale picture detail boosting technique that successfully enhances the dataset. Lastly, we provide a dataset (LPDE-4K) with a variety of LP kinds (e.g., color, country, lighting, distortion) in order to impartially assess the performance of LP corner recognition techniques. Our method performs better than other state-of-the-art methods in terms of increased accuracy and reduced computational cost when tested on a variety of datasets.

Keywords:

Convolutional neural networks, license plate (LP) detection, and unrestricted situations.

INTRODUCTION

Automatic License Plate Recognition (ALPR) technology has advanced significantly in recent years and is now commonly used in a variety of settings, including underground as well as outdoor parking lots. Numerous studies have concentrated on high-precision LP detection in an effort to increase the accuracy of LP recognition. There are numerous LP detection techniques that have demonstrated excellent accuracy in controlled settings [1, 2, 3]. However, because of the variety of environmental factors, LP detection in unconstrained scenarios is still difficult. Two-stage networks (LP detection and LP correction) and single-stage networks are the two main LP detection architectures [2], [3], [4]. Ensuring full LP information from a scene and correcting the extracted LP information are the main goals of these LP detection systems. An effective framework for LP detection ought to be able to identify LPs in situations with no restrictions. In this work, we suggest a two-component framework for LP detection that includes This function is especially helpful in minimizing the discrepancy between expected and real corner points. To enhance the dataset, we suggest a stochastic multi-scale method for

enhancing image detail. We also present the LP corner points evaluation approach (LP-NME) and an LP corner detection evaluation dataset as an extra contribution.

To summarize, we have made the following contributions: (1) DPOD-NET, a lightweight Deformation Planar Object Detection Network, can fix a variety of issues detected LP corner points and distorted LPs. By reorienting the license plate to a frontal-parallel view through the LP corners, we can lessen the perspective distortion.

(2) A new Wing loss function for LP detection that is differentiable at zero and exhibits better convergence when the error between the true and predicted values is minimal, in contrast to the L1 function. Results from experiments show that this approach outperforms others while preserving network efficiency in terms of model accuracy and other factors. (3) A stochastic multi-scale image detail enhancement method called "Data Augmentation" is suggested. The approach's benefits in enhancing image quality and detail sharpness are shown by experimental findings, and our method successfully expanded the dataset. (4) LPDE-4K, an evaluation dataset for LP detection, which includes pictures taken in a variety of unrestricted situations. Sections IV and V detail our evaluation approach for LP corner identification (LP-NME), which we propose as an extra contribution by attempting to objectify the LP corner points for assessment. The paper's remaining portions are arranged as follows. An summary of the work on LP detection is given in Section II. The strategy put out in this study is explained in Section III. dataset, we suggest a stochastic multi-scale method for enhancing image detail. We also present the LP corner points evaluation approach (LP-NME) and an LP corner detection evaluation dataset as an extra contribution of a lightweight Deformation Planar Object Detection Network (DPOD-NET) and an effective YOLOv5 vehicle detection network. While DPOD-NET is utilized for LP detection and correction, YOLOv5 is used to detect cars in unconstrained settings, which reduces the range of LP detection. Thus, the goal of our research is to create a Deformation Planar Object Detection network that is lightweight and capable of detecting vehicle LPs. Additionally, we employ a novel LPWing loss function to enhance the network training procedure in order to solve the issue of bringing the prediction of the LP corner point closer to the real value. The LPDE-4K dataset we gathered is explained in Section IV, and a number of experimental findings are shown in Section V. Lastly, we offer a methodical synopsis and future work in Section VI

OBJECTIVES

A. Detection of LP Significant advancements have been made in computer vision in recent years, especially in the area of object detection, and acknowledgment [5, 6, 7]. Nonetheless, the majority of suggested methods in the field of license plate (LP) recognition continue to rely on conventional image detection techniques, including region-based, pixel-to-pixel, and colour-based approaches [8–9]. Although these methods have unquestionably improved LP detection technology, their accuracy is limited in unrestricted situations. Convolutional neural networks (CNNs) have recently been used by more academics to address the LP detection problem [2], [10]. Numerous innovative methods have been put forth, including the LP detection algorithm developed by Hsu et al. [11] and based on YOLO [12] and YOLOv2 [13], which showed good precision and recall performance. However, more research is required to address the low performance for small item detection in unconstrained circumstances and the absence of an objective assessment of bounding boxes. To extract LPs, Li and Shen [14] employed a cascade structure with a deep neural network (DNN). For the last assignment, they subsequently suggested an end-to-end network [15]. A CNN model that was exclusively trained on artificial LP images was presented by Björklund et al. [16]. They converted three distinct datasets of LP images from different nations into frontal-parallel views and trained an LP corner detector for a regression model. Despite having very good experimental outcomes, retraining was necessary for various LP kinds. In order to identify potential LPs in scenarios, Selmi et al. [2] used picture preprocessing methods as edge detection and image morphological processing. Then, for LP detection and recognition, respectively, two distinct CNN networks were used. This method is susceptible to distorted LP and uneven lighting, too. Two YOLO-Based detection networks for spinning LPs were proposed by Xie et al. [17]. One network was used to locate the LP's region, while the other network captured a rotating rectangular box. The author's work solely took into account plane rotation, and the experimental outcome did not match the LP's original shape. Laroca et al. [18] lever - An ALPR system that was built using aged YOLO to detect LPs performed better on the SSIG dataset. The use of these models for license plate (LP) detection has grown in popularity due to recent developments in object detection algorithms. However, the traditional detection method can reduce the robustness of LP detection and is not appropriate for

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

LP correction, as it only yields a rectangular box with two points. Because it can precisely correct distorted LP, LP identification based on LP corner points has grown in popularity as a solution to this problem.

By extracting LP corner points, Silva and Jung [19] introduced a novel convolutional neural network (WPOD-NET) that can correct the deformed plane of LP. WPOD-Net has performed better on datasets like the AOLP, NET. In order to further enhance WPOD-NET's performance, Silva et al. [20] suggested an enhanced LP correction network (IWPOD-NET), which adds network parameters to boost performance on AOLP. Nevertheless, the running speed drops below WPOD-NET as the model gets bigger. Furthermore, the authors limited the LP detection performance by using YOLOv3 [21] as a vehicle detector. Thus, automatic license plate recognition (ALPR) systems can function better when LP corner points are used for LP rectification.

Actually, a lot of research has been done on facial recognition using similar methods. For instance, face identification comes first, followed by face alignment using face key points [22], [23]. A face detection method based on Yolov5 was proposed by Qi et al. [24], who performed key point regression using the Wing loss function [25]. Notably, the technique can also be applied to the detection of LP corner points. Nevertheless, there isn't an assessment method for LP corner point detection at the moment. To solve this problem, we suggest the LP-NME evaluation method in this study. The experimental findings show how successful the LP-NME evaluation method is. LP DETECTION INFORMATION Many datasets are available for LP detection model testing and training, including popular ones such as Open ALPR (available at <https://github.com/openalpr/benchmarks>), AOLP [11], SSIG [27], CD-HARD [19], UFPR [18], and CCPD [28]. It's important to note, though, that only CCPD of these databases has LPs with four corners; additionally, it only contains Chinese LPs and only uses one colour. Regrettably, there is currently no complete dataset available for assessing LP crucial corner locations. We have gathered the medium-sized LP corner points dataset, LPDE-4K, in order to address this problem. Our dataset includes difficult photos taken in unrestricted conditions, like those with motion blur, uneven lighting, huge bevel angles, and poor resolution. It also includes photos from a variety of sources, including on-the-spot collections. Additionally, we featured pictures of unrestricted electric trucks, buses, and motorcyclists. Our collection contains LP hues that aren't only blue; they also include white and yellow. Furthermore, the LP kinds in our collection include American, British, Brazilian, Chinese, Taiwanese, and European LPs.

METHODOLOGY

Fig. 1 depicts our license plate (LP) detection pipeline in its entirety. There are primarily two modules in this pipeline:

(1) Identification of Vehicles.

(2) DPOD-NET, or Deformation Planar Object Detection Network. The lightweight Deformation Planar Object Detection Network (DPOD-NET) and the vehicle detection technique will be thoroughly introduced in the part that follows.

A. DETECTION OF VEHICLES

Many exceptional object detectors are offered, taking into account aspects like managing various capture scenarios. For this article, we have chosen Yolov5s6 as the vehicle detector based on scenarios, vehicle types, and operating efficiency. Before running DPOD-NET, we scale the input image dimensions to account for the potential for large differences in image dimensions produced by the Yolov5s6 model, trying to incorporate as much of the license plate as possible.

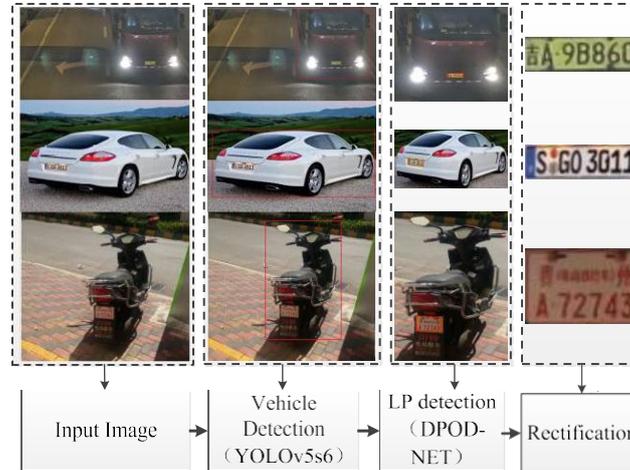


FIGURE 1. Illustration of the proposed pipeline

as much of an image as you can. When compared to the cropped width of the car, the width of the license plate changes very little in most frontal or back viewpoints. Consequently, The aspect ratio of the vehicle's bounding box is typically near 1. However, because the vehicle's side view is included in the picture, the aspect ratio of the vehicle bounding box tends to be smaller in oblique views. We utilized scaling factor [20] F_{sc} to modify the license plate picture based on the aforementioned study and related datasets (which will be covered in Section V-A). F_{sc} can be found using the formula

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

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)

Fig. 2 shows the steps involved in using DPOD-NET. The vehicle detector's pictures are fed into DPOD-NET, which creates a seven-channel feature map (r1-r7). The probability map for differentiating between license plates and non-license plates is represented by channel r1, and the feature maps for affine transformation parameters are found in channels r2-r7. We define a fixed-sized virtual square with the center cell coordinates (m, n) in order to retrieve distorted license plates (LP). We use the regression parameters to create an affine matrix and transform the square into the LP region if the object probability of that specific cell exceeds the specified detection threshold.

(1) Network Architecture

The full architecture of our network, which is mostly composed of modular CBH convolution layers, is shown in Fig. 3. DPOD block layers, and CB convolution layers. Our network's convolution filters range in number from 16 to 128 and have sizes of 3×3 and 1×1 . Throughout the network, we employ the Hard-Swish activation function [29]. Throughout the network, the stride of the CB convolution layer is set to 2. Two branches make up the last layer of our network: one is intended for LP localization and returns the LP probability, while the other is in charge of extracting six affinity transformation parameters that are triggered by the sigmoid function

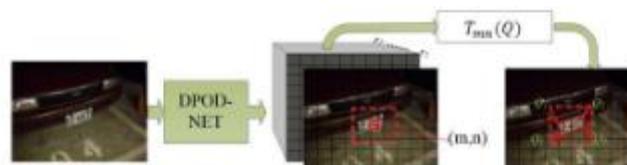


FIGURE 2. License plate detection and rectification process.

(2).DPOD Block

The creation of lightweight networks like ShuffleNet v1 has attracted a lot of attention in recent years. [29], [30], and [31]. ShuffleNet v1 tackles problems with weak representation between channel groups and information flow. We first apply a channel shuffling operation to the input features in the DPOD Block of our network, and then we apply a layering operation. For one of the branch feature maps, we employ a computationally efficient 3×3 universal convolution module and a 1×1 convolution module [32], after which we carry out information stacking. Lastly, we do another channel shuffling operation after concatenating the data from both branches. Figure 3(b) shows an illustration of the DPOD Block.

(3). Block of Transformer Encoders

We have added a transformer encoder block to our network in accordance with the vision transformer concept [33], [34]. improve its functionality. The block can extract both global and contextual information, as seen in Fig. 3(c). A multi-attention layer and a fully connected layer (MLP) are the two sub-layers that make up each transformer encoder.

To enhance the network's overall performance, residual connections are used between each sub-layer. Our model can efficiently capture a variety of local information by leveraging the transformer encoder block, and it can also investigate the possibilities of self-attentive mechanisms [35].

C. THE FUNCTION OF LOSS

(1).The normalizing function and affine transform According to Silva and Jung's description [19], the definition of the The following is an expression for LP corner points. Define the annotated LP corner points as $P_i = [x_i, y_i]$, beginning with the top left LP corner points. T, define $Q_1 = [-0.5, -0.5]$ appropriately for $i = 1, \dots, 4$. $Q_2 = [0.5, -0.5]$ TT, Q_3 equals $[0.5, 0.5]T$, $Q_4 = [-0.5, 0.5]T$ as the unit square's corner points, with the origin at its center. The network output feature map is made up of $M \times N \times$, assuming that the RGB image is represented as $I \in \mathbb{R}^3 \times H \times W$ and that the network step $N_s = 16$. $M = H/N_s$ and $N = W/N_s$ for 8 volumes. The affine transform T_{mn} can be defined as follows: the network generates seven estimates for each point cell (m, n) in the feature map, with the first estimate (r_1) representing the probability of object/non-object and the remaining values (r_2 to r_7) being used to parameter values of the affine transform for LP correction.

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

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 in order to match the network output's resolution. This is achieved by the use of a normalization function.

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

where the side of the imaginary square is represented by the scaling parameter α . The average point between the maximum and smallest LP dimensions in the enhanced training data, divided by N_s , is what we assign to $\alpha = 7.75$.

(2). LPWing loss function

Popular loss functions like L2, L1, and smooth-L1 are frequently used for key point regression applications [7]. As an example, the The L2 loss function is employed in the popular MTCNN [22] model for face key points regression. These loss functions, however, are typically less susceptible to minor errors. New loss functions that can enhance the network's training capability in small-to-medium ranges close to zero have been developed as a result of this limitation.

Drawing inspiration from recent studies [22], [23], [24], [25], and [36], we suggest the LPWing loss function as a remedy. The following is the definition of the LPWing loss function.

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

where x_k represents the discrepancy between the warped plane's corner points and the annotated LP corner points. The positive parameter μ is set to the interval $[-\mu, \mu]$ and is mostly utilized for Extremely extremely low values. We utilize the \ln function to limit the curvature of the nonlinear region by parameter ϵ for small values of x ($x \in [\pm\mu, \pm\omega]$), where $K = \omega / (\epsilon + \mu)$ and $A = \omega * \ln(1 + \mu / \epsilon)$. $C = \omega + \omega * \ln((\epsilon + \mu) / (\epsilon - \omega)) - \omega * \mu / 2 * (\epsilon - \mu)$ and $K * \mu / 2$ are used to modify the linear and nonlinear smoothing constants. It is crucial to remember that setting the value of ω too high or too low can cause network instability and an explosion issue. Furthermore, To guarantee the gradient's continuation, we employ x_{2k} in the zero-entry portion. The LPWing loss is displayed in Fig. 4. function for various parameter configurations and loss function curves. Although it is not set to zero in the interval $[-\mu, \mu]$, the LPWing loss function functions similarly to the R-Wing loss [36]. Figure 5(a) displays several loss functions, while Figure 5(b) displays the gradient curves that correspond to them. For L1, the step size is disproportionately impacted by significant errors, while the gradient magnitude remains constant across all sites. However, L2 keeps the step size constant while the gradient is affected by significant mistakes. Although Smooth L1 is a hybrid of L1 and L2, it has trouble fixing very minor mistakes.

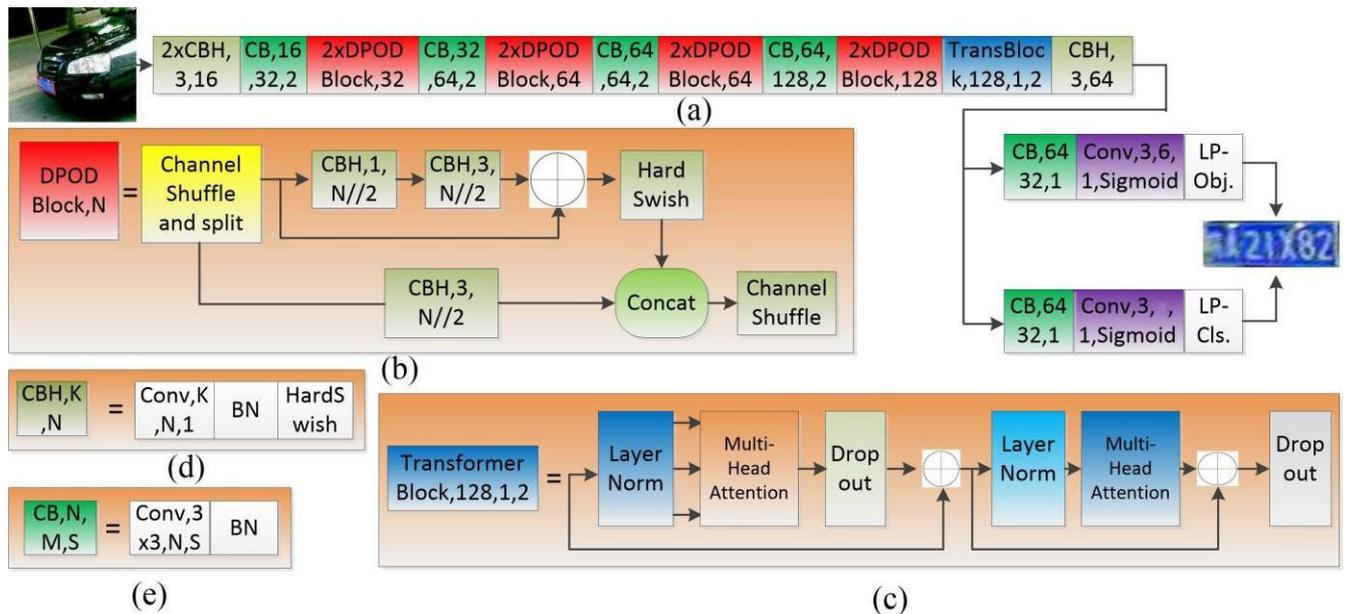


Figure 3: DPOD-NET Overview 208 x 208 is the size of the input image. (a) DPOD-NET: This is DPOD-NET's general architecture. (b) DPOD Block: DPOD-NET uses this particular block. The number of channels is represented by the character "N." (c) Architecture of Transformer Encoders: A feed-forward neural network (MLP) and a multi-head attention block make up this component's two primary building pieces. Convolution Layer CBH (d): This layer makes use of a three-by-three filter with a stride of one. (e) CB Convolution Layer: This layer employs a stride of 2 and a 3 x 3 filter.

The $\ln x$ function, with a gradient of $1/x$, can be used to account for the impact of small errors. Nevertheless, as x gets very small, the gradient Change takes on significance. Therefore, we use L2 loss when $|x_k| \leq \mu$. The LPWing loss function's gradient is smooth and continuous, as seen in Fig. 5(b), which is advantageous for training data that is near zero.

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

When ω was set to 6 and ϵ to 2, the model reached its maximum level of efficacy, according to our experiment, where we set $\mu = 0.1$. Please refer to Section V, where we provide comprehensive ablation studies, for a more thorough examination of the effects of various parameter settings.

The mapping relationship between the four corners of the square and the four corner points of the license plate is shown in Fig. 2 (shown by red arrows). The inaccuracy between the four points of the square and the four corner points of the license plate is determined using the LPWing loss function. If $x_k = T_{mn}(Q_k P) - A_{mn}(P_k)$, then the four points' total loss can be written as

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

(3). Loss function for LP classification

We use focal loss [37], which may be written as follows, to forecast whether there is LP at (m, n) .

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

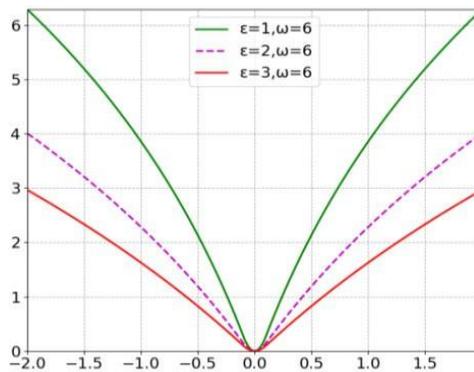


FIGURE 4. Curves of the LPWing loss functions, where $\omega = 6$.

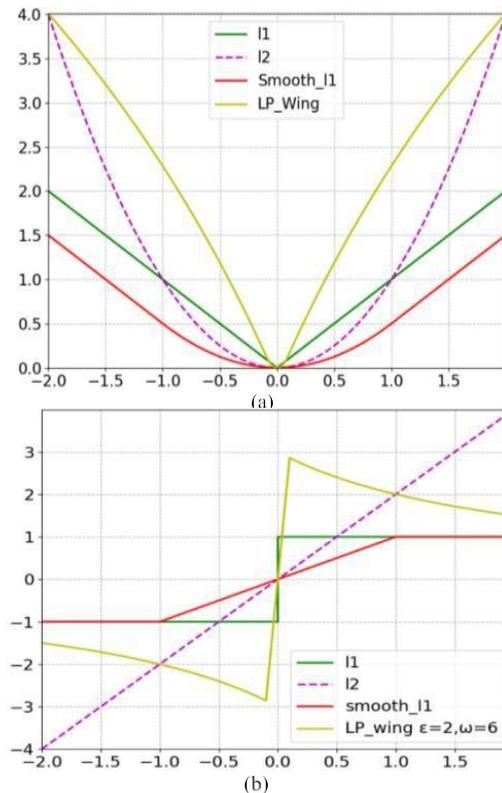
$P \in [1, 0]$ for the probability of LP, $\alpha \in [1, 0]$ for the weight factor, $\alpha = 0.25$, $y \in [+1, -1]$ for the ground truth category, and γ is known as the focusing parameter, $\gamma = 2$. A combination of the terms from (5) and (6) yields the final loss function:

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

$wc = 0.5, wl = 0.5$.

D. TRAINING AND DATA AUGMENTATION

We use a number of data augmentation techniques to help the trained model better adapt to different complex settings methods. First, we make geometric adjustments, including changing the photos' size and perspective. Next, we make advantage of color changes by altering the HSV color space at random with a $p=100\%$ chance. Furthermore, we use a Gaussian fuzzy technique with a $p=15\%$ probability, where random selection is used to determine the degree of fuzziness σ , which falls between 0 and 0.1N. Furthermore, in order to improve the training set image detail and enhance the dataset, we provide a stochastic image multi-scale detail boosting technique. This procedure is explained in more detail in the next section.



The L2, L1, smooth L1, and LPWing loss function curves are shown in Figure 5(a).

(b) The derivative curve for various loss functions. $\omega = 6$ and $\varepsilon = 2$ are the parameters we set.

The training set for this study includes a variety of car models and license plates (LPs) from several nations, which leads to LPs with variable shapes. To address this challenge, we use the strategy Silva et al. [20] suggested. On the other hand, we define the aspect ratio as $ar \in [1.25, 2.5]$ for LPs with close length and breadth (like electric motorcycles) and as $ar \in [2.5, 5]$ for LPs with significant length and width disparities. Then, to create LP samples with different 3D rotations, we select angles of ± 55 , ± 75 , and ± 75 . Lastly, the input training image and gaze locations are distorted, and a random background is added to create an augmented image.

We have enhanced Kim et al.'s multi-scale picture detail boosting technique to significantly enhance the quality of the converted data [38]. In particular, we generate four alternative versions of the globally sourced image I, each with varying degrees of blur, using Gaussian kernels. We can use the four distinct blurred images that are produced by this procedure to increase the original image's detail at different scales.

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

where the Gaussian kernels G_1 , G_2 , G_3 , and G_4 have the standard deviations $\sigma_1 = 1.0$, $\sigma_2 = 2.0$, $\sigma_3 = 4.0$, and $\sigma_4 = 6.0$, respectively. We take out every detail by

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

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

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

where the values of $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6,$ and ω_7 are set to 0.5, 0.5, 0.25, 0.5, 0.5, 0.25, and 0.125, respectively. Be aware that simple superposition may cause the image to appear oversaturated. applied. In order to balance the image pixels and prevent image oversaturation, we employ the hyperbolic tangent function technique [39]. It is evident from Fig. 6 that our method performs better than the original method. By setting the Gaussian kernel size (ksize) as a random value, our data augmentation method allows for the creation of various detail-boosted images. To a certain degree, this procedure expands the quantity of data in the initial training set. The outcomes of our tests for using different Gaussian kernel sizes are shown in Fig. 7. Interestingly, the detail-boosted image is not visible when ksize=1. However, the visual detail significantly improves as the ksize grows. When training the model, we set the ksize value to a list [1, 1, 3, 5, 7, 9, 11], from which we choose a value at random to represent the current Gaussian kernel's size. As illustrated in Fig. 8, which displays the impact of the augmentation pipeline on a sample image, the variety and quality of the data are improved when compared to the original dataset through the application of data augmentation approaches discussed above. We produced a tiny training set of 842 photos in order to train our model. The collection includes 100 pictures of trucks, 124 pictures of electric motorcycles, and 100 pictures of automobiles, buses, and vehicles having different colored license plates. The size of the input image is 208 by 208. We used the adamax optimizer [40] with a batch size of 64 and an initial learning rate of 0.01 during the model training phase. In a training procedure, the setting of the learning rate and iteration count can be written as

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

where N is the number of iterations and LR is the learning rate. As stated in Section V, we ultimately selected the model with the best performance metrics.

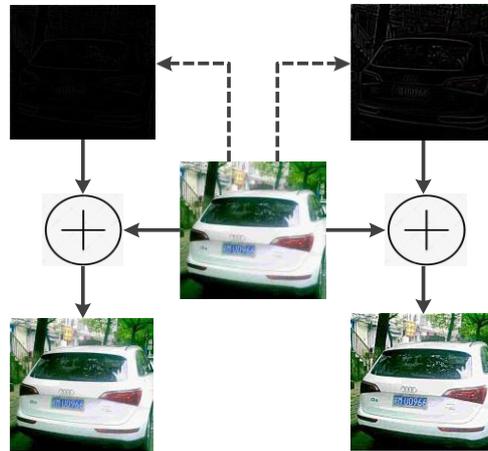


FIGURE 6. Comparison of the multi-scale image detail boosting effect, using a Gaussian kernel size of 5. The original image, obtained from CCPD, is shown in the center of the figure. The top-left corner of the figure displays the image detail extracted using the original approach, while the top-right 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.

LP CORNER POINTS DETECTION AND EVALUATION DATASET

The primary datasets used for LP detection include AOLP [11], SSIG [27], OpenALPR (accessible at <https://github.com/openalpr/benchmarks>), CD-HARD [19], UFPR [18], and CCPD [28]. These datasets contain a small number of LP types. CCPD, for instance, comprises many car LPs, a few bus LPs, and just one LP color. AOLP consists solely of car LPs. As the global population of electric vehicles grows, it becomes more crucial to identify a significant number of electric vehicle LPs. Consequently, we built a dataset of multiple vehicles that contains 4000 images. The dataset includes various vehicle types such as electric motorcycles, cars, trucks, and buses. The LP colors consist of blue white, yellow and red. The dataset also includes LPs from various countries including China, Brazil, Iran, the USA, Taiwan, Europe, and the UK. Additionally, our dataset indicates the four corners of LPs, making it suitable for the detection and assessment of LPs. In constructing the LPDE-4K dataset, we considered multiple factors that influence image quality, such as lighting, angle, distance, and intricate backgrounds. Consequently, the dataset we propose is very difficult in every regard. Most of the images in LPDE-4K come from unconstrained scenarios with different image tilts, distances, lighting conditions, and blurring. This variability complicates the task of LP detection. In addition, our dataset encompasses a variety of real-world situations in which LP detection might be necessary.

RESULTS AND DISCUSSION

All our code implementations are grounded in the PyTorch learning framework, and we performed our experiments on an Intel Core i5-9400F and NVIDIA GTX1070TI GPU Processor. We have empirically established the acceptable thresholds for two networks in our proposed approach, which are in the pipeline. In particular, we established the threshold for vehicle detection with YOLOv5s6 at 0.35 and the threshold for license plate (LP) detection with DPOD-NET at 0.45. Our LP detection network detects the area produced by the vehicle detector YOLOv5s6, which is important to keep in mind.

A. DATASET

The goal of our experiment reported in this paper is to test how well and reliably an algorithm works for LP detection. For the assessment for the performance, we utilized four datasets: CCPD, AOLP, OPENALOP (accessible at: <https://github.com/openalpr/benchmarks>), and LPDE-4K. (1) AOLP [11] is divided into three categories: Tilt LP, Front LP, and General LP. These images are categorized into three subsets: AC (681 samples), LE (757 samples), and RP (611 samples). We performed a re-annotation

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

of the corner points of all LPs in the AOLP dataset as part of our experiments. (2)CCPD [28] constitutes a substantial, varied, and meticulously annotated open-source dataset of Chinese city LPs. The data for this dataset were gathered in the He Fei car park between 7:30 a.m. and 10:00 p.m. The LP images encompass a range of intricate environments, such as blur, tilt, rain, snow, and additional conditions. This dataset comprises more than 250,000 LP images of Chinese cities along with annotations for number plate detection and recognition.



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-boostered image.



FIGURE 8. The ultimate augmentation effect of the identical image.

(3) OpenALPR (<https://github.com/CarltonSemple/openalpr>) is a library for automatic license plate recognition that is open-source and includes a dataset of more than 800 license plates, comprising Sub-databases for the

EU, BR, and US. We assembled a dataset of 60,000 vehicle images, comprising cars, buses, motorbikes, and trucks, to train the YOLOv5s6 network. The vehicle images were sourced from various datasets, including CCDP [28], BDD100K (accessible at <https://bdd-data.berkeley.edu>), COCO2017 (accessible at <https://cocodataset.org/#home>), VOC2012 (accessible at <http://host.robots.ox.ac.uk/pasc-al/VOC/voc2012/>), AOLP [11], SSIG [27], and data we compiled ourselves.

B. EVALUATION METRICS

We employed standard precision (PR), recall (RE), and F-Measure (F-M.) evaluation metrics to ensure the accuracy of our model. protocols for evaluating how well our model performs on the AOLP dataset. Additionally, we utilized the same dataset to assess how effective the LPWing loss function is. To assess the accuracy performance of our model, we employed the CCPD benchmark. The performance of face key points detection is often assessed using the Normalization Mean Error (NME) [26], [36]. To the best of our knowledge, a comparable method does not exist in LP corner point evaluation. We proposed an enhanced version of NME, called LP-NME, which serves as an evaluation method in our experiment. The definition of the LP-NME for each LP image is as follows

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

where P and \hat{P} denote the annotated and predicted LP corner points for each image, respectively. Furthermore, N represents the total count of LP corner points. calculate the normalization factor for each image. For the AOLP dataset, we used the diagonal measurement of the smallest enclosing rectangle around the LP as the normalization factor. Fig. 9 shows an illustration of the smallest rectangular box, which is depicted in blue. The LP-NME calculation serves to ascertain the mean of the relative error between predicted and actual corner points. It is essential to emphasize that choosing a suitable normalization factor, represented by d , is vital for this calculation.

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	X	1
CCPD(2018)[28]	250K	2	1	1	2/4	1
OpenALPR	800	3	1	2	X	1
LPDE-4K	4K	4	4	5	4	1

TABLE 1. Comparison of key attributes in the license plate detection dataset: images: the dataset's total number of images. kinds of vehicles: the hue of the license plates (LPS) for various kinds of vehicles. LP nations: LPS originating from different nations that are part of the dataset. Real scenes: occurrences of lps and non-lps found in actual world scenes. LP marking: the quantity of corner points that indicate the LPS within the images. LPS/IMAGE: the mean quantity of LPS shown in every picture.

C. LPWING LOSS FUNCTION PARAMETERS AND EVALUATION

To find the optimal parameter settings for LPWing loss function, we started by manually marking the LP corner points of the AOLP dataset. Subsequently, we examined various combinations of parameters and assessed how well they performed on the AOLP dataset. To reduce the search space for the parameters, we initialized ω and ϵ at 6 and 1, respectively, while μ was set to 0.1. We thought that if the values of ω and ϵ were set too high, it could lead to network instability, whereas if they were set too low, it could result in the gradient explosion problem.

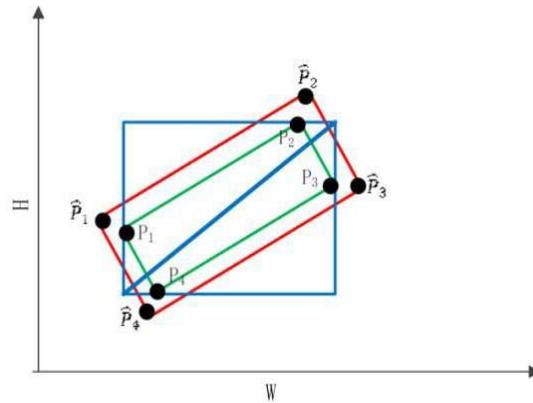


FIGURE 9. Depiction of the LP-NME employed to assess the LP corner points. The predicted LP corner points are indicated by \hat{P}_i ($i=1...4$), whereas the annotated LP corner points are represented by P_i ($i=1...4$). The blue rectangular box has a diagonal length represented by d .

$\epsilon \backslash \omega$	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

TABLE 2. Assessment of lpwing loss with varying parameter ω configurations.

We assessed various combinations of ω and ϵ , training DPOD-NET for 50k iterations with each combination and calculating the average LP-NME values on the AOLP dataset for all combinations. The TABLE 3 presents the findings of this experiment, demonstrating that our LPWing converges more quickly when $\omega = 6$ and $\epsilon = 2$. After determining the best values for ω and ϵ , we set them and analyzed the performance of various loss functions (L1, L2, Smooth-L1, and LPWing). TABLE 3 presents the results, demonstrating that the LPWing loss function yielded the lowest LP-NME value, which signifies that the predicted value was nearest to the actual value. As demonstrated in TABLE 4, LPWing consistently produces smaller LP-NME values than the L1 loss function across different models. Ultimately, we contrasted the performance of various loss functions for various network frameworks in TABLE 5.

$\epsilon \backslash \omega$	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

TABLE 3. The test results performed by DPOD-NET indicate that when the values of ω and ϵ are 6 and 2, respectively, the average LP-NME (AVG.) of different loss functions was computed. The results show that the lpwing loss function performed better than the other loss functions



FIG. 10. when contrasting our method with WPOD-NET [19] and IWPOD-NET [20], the red box denotes the detection outcome, whereas the green box signifies the actual box. Values of the corrected LP and LP-NME can be located in the lower left corner of every image. This comparison utilizes images from the LPDE-4K and ALOP datasets.

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

In this section, we present the performance of our proposed approach in relation to other cutting-edge methods. To assess our method, we used precision (PE) and recall (RE) rates and F-Measure (F-M.), which are frequently utilized in LP detection. Due to the absence of an objective evaluation standard for LP corner point detection, we also employed LP-NME for performance assessment, as it has seen widespread use in LP detection in recent years. The experimental findings show that LP-NME can be trusted for detecting LP corner points.

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

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

TABLE 4. LP-NME values for the L1 and LPwing loss functions across different models



FIGURE 11. Partial image samples, showcasing a variety of objects. The top row contains two images from the CCDP dataset, while the rest are from the LPDE-4K dataset. The red box appearing in every image

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

denotes the outcome of object detection, whereas the green box represents the ground truth bounding box. The values of the LP and LP-NME metrics after correction are shown in the lower left corner of each image.

In the ALOP sub-dataset experiment, we solely compared the performance of our DPOD-NET without using our trained YOLOv5s6 vehicle detector. We performed a re-annotation of the ALOP dataset encompassing all LPs. The results in TABLE 5 indicate that our method surpasses other state-of-the-art methods. On the AC subset, DPOD-NET achieves a precision of 99.9, surpassing that of IWPOD. Moreover, the recall and F-M. values exceed those reported by Björklund et al. [16]. The average LP-NME on the LE and RP subsets is smaller, while precision is higher, suggesting that DPOD-NET performs better regarding deformed planar object detection. We chose five subsets from the CCPD dataset for our experiment and took into account the number of samples in each subset. Detection accuracy results compared with existing LP detection approaches are shown in TABLE 6. The outcomes suggest that DPOD-NET attains the greatest accuracy in detecting LP corner points on the DB. subset, with a detection accuracy reaching 99.1 using YOLOv5s6-DPOD-NET, exceeding other methods. Despite DPOD-NET achieving the highest accuracy on the Rot., Tilt., and Weath. subsets, its accuracy on the Chall. subset is slightly lower than IWPOD-NET's, recording a detection accuracy of 92.7. Nevertheless, when considering all the other methods, YOLOv5s6-DPOD-NET achieves the best accuracy. It is noteworthy that the YOLOv5s6 vehicle detector simplifies the scenarios and generates results that are more favorable for LP detection with DPOD-NET. For the vehicle detector, we utilized 60,000 vehicle images from various datasets for training. In contrast, DPOD-NET was trained on just 842 images, of which 420 were sourced from the CCPD dataset. Additionally, TABLE 7 provides a summary of the comprehensive detection performance of our approach on the ALOP, OPENALOP, and LPDE-4K datasets. Our method achieves superior outcomes on the ALOP dataset, with LP-NME values of 0.220 and 0.180, respectively. This indicates that the LP corner points are nearer to the actual values. In all metrics except precision, DPOD-NET outperforms IWPOD-NET on the OPENALOP and LPDE-4K datasets. Ultimately, by integrating the YOLOv5s6 vehicle detector into With DPOD-NET, we can decrease LP-NME by 0.08 on both the OPENALOP and LPDE-4K datasets, as well as enhance other metrics. All in all, these findings show that our method is the best fit for LP detection in unrestricted scenes.

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

Approach(Year) Subset	DB.(190k).	Rot.(10k).	Tilt.(30k).	Weath.(10k).	Chall.(50k).
Cascade (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
RPNNet (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 6. The accuracy of LP detection (IN %) for CCPD.

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

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

Model(Year)	Model size (MB)	Parameters (MB)	Floating point Operations (FLOPs)	Detection speed (ms/Image)		
				GTXT1070Ti	I5-10300H	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

TABLE 8. Information on the complexity of the models (input image size: 208) × 208.

Electric motorcycles are shown in Fig. 10 with LP colors of white and yellow; their shape is nearly square. With the smallest LP-NME value, our method produces the best outcomes.

while IWPOD-NET identifies a false positive in the first row of electric motorcycle LP detection. In the last two rows of images, the LP color appears white, and the LP is distorted. Our method of detection yields superior results and offers greater precision in the location of the vehicle LP, as demonstrated by the minimum LP-NME value. Fig. 11 shows a selection of qualitative results to illustrate the actual outcomes of YOLOv5s6-DPOD-NET. The first two images come from the CCDP dataset, whereas the others are sourced from LPDE-4K.

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 find that DPOD-NET has a lower number of parameters than IWPOD-NET, leading to a 39% decrease in FLOPs. Furthermore, the model size of DPOD-NET is only

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

4.43M. Additionally, we assessed the mean duration needed to process one image from the ALOP dataset.

CONCLUSION

In this study, we concentrated on examining the identification of deformed license plates (LPs) in unconstrained situations. In order to accomplish this, we put forward a successful deformed planar object detection network (DPOD-NET), featuring multiple essential contributions. To begin with, we created a network module. We then put forward a plane distortion detection network that identifies corner points of distorted LPs to correct them. Our network was trained on just 842 images, yet experiments demonstrated its strong generalization ability across various unconstrained scenarios. Additionally, our network showed it could identify different kinds of LPs. Secondly, we implemented the LPWing loss function to improve the training effectiveness of the loss at positions near the zero point during network training.

This contributed to reducing the discrepancy between the predicted and actual values. We also suggested the image random multi-scale approach to enhance the dataset's quality. We provided further information on the boosting algorithm applied in data augmentation. At last, we established a new standard for assessing LP corner points, named LPDE-4K. Our suggested method for detecting license plates (LPs) shows performance that meets or exceeds that of leading methods across all evaluation metrics, even in difficult, unconstrained situations, according to the experimental findings. It is worth mentioning that our method surpasses all other approaches when YOLOv5s6 serves as the vehicle detector. Besides accurately identifying LPs, our method can proficiently manage plane distortion correction for electric motorcycles, cars, trucks, and buses.

In future work, we will design an efficient end-to-end network for license plate detection and correction that can be used on various embedded devices or vehicular edge computing devices [41], [42], including 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. B. 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.

IJETRM

International Journal of Engineering Technology Research & Management

Published By:

<https://www.ijetrm.com/>

- [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," *arXiv preprint*, arXiv:1601.05610, 2016.
- [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.