A Flexible Approach for Automatic License Plate Recognition in Unconstrained Scenarios

Sergio M. Silva and Cláudio Rosito Jung, *Senior Member, IEEE*

***Abstract*— Automatic License Plate Recognition is a crucial task for several applications related to Intelligent Transportation Systems, from access control to traffic monitoring. Most existing approaches are focused on a specific setup (e.g., toll control) or a single license plate (LP) region (e.g., European, US, Brazilian, Taiwanese, etc.), which limits their application. This work proposes a complete ALPR system focusing on unconstrained capture scenarios, where the LP might be considerably distorted due to oblique views. We present an Improved Warped Planar Object Detection Network (IWPOD-NET) that is able to detect the four corners of an LP in a variety of conditions, so that it can be warped to a fronto-parallel view and alleviate perspective-related distortions. Given the rectified LP, we test two different Optical Character Recognition (OCR) methods based on object detection. Our experimental results show that the proposed detector is competitive with state-of-the-art (SOTA) methods using a very limited training set. Regarding the full ALPR results, our method achieves top-scoring results for several datasets that include a variety of capture conditions and vehicle types (in particular, motorcycles).**

***Index Terms*— License plate detection, license plate recognition, perspective correction, deep learning.**

I. Introduction

SEVERAL traffic-related applications, such as detection of stolen vehicles, toll control, and parking lot access validation, involve vehicle identification, which is performed by Automatic License Plate Recognition (ALPR) systems. The recent advances in Parallel Processing and Deep Learning (DL) have contributed to improving many computer vision tasks, such as Object Detection/Recognition and Optical Character Recognition (OCR), which clearly benefit ALPR systems. In fact, deep Convolutional Neural Networks (CNNs) have been the leading machine learning technique applied for vehicle and license plate (LP) detection [1]–[10]. Along with academic papers, several commercial ALPR systems also benefit from the rise of DL methods. They are usually allocated in huge data-centers and work through web-services, being able to process thousands to millions of images per day and be constantly improved. As examples of these systems, we can mention Sighthound (https://www.sighthound.com/) and the commercial version of OpenALPR (http://www.openalpr.com/)).

Despite the considerable advances in the state-of-the-art (SOTA), devising a generic ALPR system that is able to handle photometric variations (variety of colors and different illumination conditions, shadows, image noise) and geometric variations (LPs with different shapes and acquired at a variety of viewpoints) is still a challenge.

In this work, we present an extended version of our conference paper [11] that is capable of dealing with a variety of scenarios and camera setups. The main improvement was the re-design of the LP detection network, with the inclusion of specialized layers to separately learn weights for both classification and localization tasks. The new network improved the results of the original version for car LPs, and now can also handle motorcycle LPs, which is overlooked by most existing ALPR approaches. As additional improvements, we present post-processing strategies than can be coupled to any detector-based license plate recognition module, and fine-tuned an object detector to identify cars (including buses and trucks) and motorcycles as output classes. We show that our modifications improve the results over [11] by a good margin, leading to SOTA results in some publicly available datasets.

II. Related Work

ALPR is the task of finding and recognizing license plates in images. It is commonly broken into two main tasks: license plate detection (LPD), which aims to locate the license plate in the image (and might be preceded by vehicle detection), and license plate recognition (LPR), which aims to generate the string related to the LP.

As the main contribution of this work is the introduction of an improved LPD network, we start this section by reviewing DL-based approaches for this specific subtask. Next, we move to complete ALPR DL-based systems.

A. License Plate Detection

The good compromise between accuracy and speed provided by YOLO networks [12]–[14] inspired many recent works to use similar architectures targeting real-time performance for LP detection [2], [6]–[8], [15]. A slightly modified

Manuscript received July 1, 2020; revised November 11, 2020; accepted January 19, 2021. Date of publication February 18, 2021; date of current version May 31, 2022. This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) under Grant Finance Code 001 and in part by the Brazilian Research Council (CNPq). The Associate Editor for this article was G. Mao. *(Corresponding author: Cláudio Rosito Jung.)*

Sergio M. Silva is with the Department of Computation, State University of Londrina, Londrina 86057-970, Brazil.

Cláudio Rosito Jung is with the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre 90040-060, Brazil (e-mail: crjung@inf.ufrgs.br).

Digital Object Identifier 10.1109/TITS.2021.3055946

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

version of the YOLO [12] and YOLOv2 [13] networks were used by Hsu *et al.* [6], where the authors enlarged the networks output granularity to improve the number of detections and set the probabilities for two classes (LP and background). Their network achieved a good compromise between precision and recall, but the paper lacks a detailed evaluation of the extracted bounding boxes. Moreover, it is known that YOLO networks struggle to detect small-sized objects. Thus, further evaluations over scenarios where the car is far from the camera are needed.

In [7], the authors presented a setup of two YOLO-based networks aiming to detect rotated LPs. The first network is used to find a region containing the LP, called "attention model", and the second network captures a rotated rectangular bounding-box of the LP. Nonetheless, they considered only on-plane rotations, which is not an adequate model when very oblique camera views are present. Also, as they do not present a complete ALPR system, so it is difficult to evaluate how well an OCR method would perform on the detected regions.

LPD methods using sliding window approaches or candidate filtering coupled with CNNs can also be found in the literature [4], [5], [16]. However, they tend to be computationally inefficient as a result of not sharing calculations like in modern meta-architectures for object detection such as YOLO, SSD [17] and Faster R-CNN [18].

B. Complete ALPR Methods

The works of Silva and Jung [2], [19] and Laroca *et al.* [8], [15] presented complete ALPR systems based on a series of modified YOLO networks. In [2], [19], the authors trained two distinct networks, one to detect cars and LPs jointly, and another to perform OCR, with the additional use of temporal information in [19]. Their method is very fast but limited to mostly frontal/rear views. Laroca and colleagues [8] used three networks based on the YOLO architecture for vehicle detection, LP detection, and character segmentation, and an additional network for character classification. Their work was extended in [15], where layout classification was included in the LP detection module, and a single network was used for character detection and classification. Both versions [8], [15] can handle motorcycle LPs, but also were not trained to capture strong geometric distortions.

Selmi *et al.* [16] used a series of pre-processing approaches based on morphological operators, Gaussian filtering, edge detection, and geometry analysis to find LP candidates and characters. Then, two distinct CNNs were used to (i) classify a set of LP candidates per image into one single positive sample, and (ii) to recognize the segmented characters. The method handles a single LP per image, and /according to the authors, distorted LPs and poor illumination conditions can compromise the performance.

Xu and colleagues [20] presented a large dataset (CCPD) and baseline for Chinese LPs, along with an end-to-end network that combines an LPD sub-network with the LPR module presented in [21], which requires only class-level annotations. Despite the good results shown in [20], the network was trained for Chinese LPs only (the number of detected characters was fixed to seven), which limits its application. Lee and colleagues [22] presented a joint denoising and rectification network to both obtain an approximately fronto-parallel view of the LP and improve the visual quality. They also presented a second network that segments the characters and regresses the expected number of characters in the LP, and recognition is performed using YOLOv3 [14]. Since they resize input images to 320×320, their method is adequate when the vehicle is framed in the input image.

Björklund and colleagues [9] presented a complete pipeline based on deep models trained only with synthetic LP images. More precisely, they generate synthetic LPs from templates (e.g., Italian or Taiwanese, as shown in the paper) and apply several geometric and photometric distortions, which are used to train a deep model that regresses the four corners of the LP, which is transformed to a fronto-parallel view. An LPR network based on object detection is trained in a similar manner, and post-processing steps that required additional knowledge (such as the number of characters in the LP) are applied. Despite the good results shown in [9], they train a specific model for each LP type (e.g., one model for Italian and another for Taiwanese), so that adding a new plate region requires re-training the networks.

There are also several other approaches that combine different network architectures for the different ALPR tasks, such as Mask-RCNNs in [10]. It is also worth mentioning that some approaches explore recurrent layers and Connectionist Temporal Classification (CTC) in the LPR step, which allows character recognition without localization annotations. For example, Li et. al [1] combined traditional features (such as HoG and LBP) with recurrent networks and CTC for character recognition, then used a deep feature extractor with a similar LPR module in [1], and later presented an end-to-end network that explores CTC in the final task [3].

Commercial systems are good reference points to the SOTA. Although they usually provide only partial (or none) information about their architecture, we still can use them as black boxes to evaluate the final output. Examples of commercial systems are Sighthound, OpenALPR, and Plate Recognizer (https://platerecognizer.com/).

This brief revision indicates the existence of several approaches for ALPR (or the LPD or LPR subtasks). However, most methods are focused on a single LP region, or represent the LP region as a rectangular bounding box (which might limit its application to a specific capture situation, such as vehicle relatively well-framed into the image, pose approximately frontal, etc.), as summarized in Table I. Our approach was designed to be generic w.r.t. the capture conditions and LP type, being also able to handle motorcycles. It will be described next.

III. The Proposed Method

The core of the proposed method is the introduction of an Improved Warped License Planar Object Detection network (IWPOD-NET), which is able to detect car and motorcycle LPs in a variety of capture situations and then rectify them to a fronto-parallel view. Although it can be applied directly to an input image, we suggest the use of a previous

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF REVIEWED METHODS

Work	LPD	LPR	T/P/S/L*	Box Type
Li & Chen[1]	✓	✓	✓✓✗✓	Rotated
Silva & Jung[2]	✓	✓	✗✓✓✗	Rectangular
Li et al.[3]	✓	✓	✓✓✓✓	Rectangular
Kurpial et al.[4]	✓	✗	✗✗✓✓	Rectangular
Bula et al.[5]	✓	✓	✗✗✓✓	Rectangular
Hsu et al.[6]	✓	✗	✓✓✗✓	Rectangular
Xie et al.[7]	✓	✗	✓✓✗✓	Rotated
Laroca et al.[8]	✓	✓	✗✗✓✓	Rectangular
Bjorklund et al.[9]	✓	✓	✓✓✓✓	4-corners
Selmi et al.[10]	✓	✓	✓✓✓✓	Rectangular

*Tested Scenarios: (T)oll, (P)arking, (S)ecurity, (L)aw Enforcement.

module for vehicle detection, which limits the search region of IWPOD-NET. The rectified LPs are then fed to an LPR module, which detects the characters (and optionally enforces region-dependent rules), providing the LP string. We start by describing IWPOD-NET, and then tackle vehicle detection and license plate recognition.

A. License Plate Detection and Unwarping

The Warped Planar Object Detection Network (WPOD-NET) is a fully convolutional network introduced in [11] for detecting the four corners of an LP. The original formulation of WPOD-NET strongly relied on weight sharing for both classification and localization tasks, which was carried out until the last layer. Dealing also with motorcycle LPs presents additional complexity to the task, leading us to add a few parallel layers in the original architecture to better distinguish the two tasks.

As WPOD-NET, the proposed improved WPOD-NET (or IWPOD-NET) is a fully convolutional network that performs fast multiple single-class object detection in a single pass. It encodes the localization parameters as a set of six affine transformation parameters that warps a canonical square to the quadrilateral that represents the LP in the possibly distorted image. As such, we can unwarp the detected LP to a fronto-parallel projection, reducing distortions due to perspective issues.

1) Network Architecture: IWPOD-NET is formed by a sequence of convolutional layers with batch normalization (convbatch), several of them being inside residual blocks aiming to alleviate the problem of vanishing gradients and accelerate training time [23]. The size of all convolutional filters is fixed (3×3), and the number of filters varies from 16 to 128. Rectified Linear Units (ReLU) are used as activation functions throughout all intermediate layers of the network, except for the final layers that specialize in the classification and localization tasks. There are four max-pooling layers of size 2×2 and stride two that reduce the input dimensionality by a factor of 16 throughout the network.

In WPOD-NET, the final block presents two parallel convolutional layers: (i) one for inferring the object probability, activated by a softmax function, and (ii) another for regressing the affine parameters with linear activation. Hence, the weights for both classification and detection tasks are shared up until

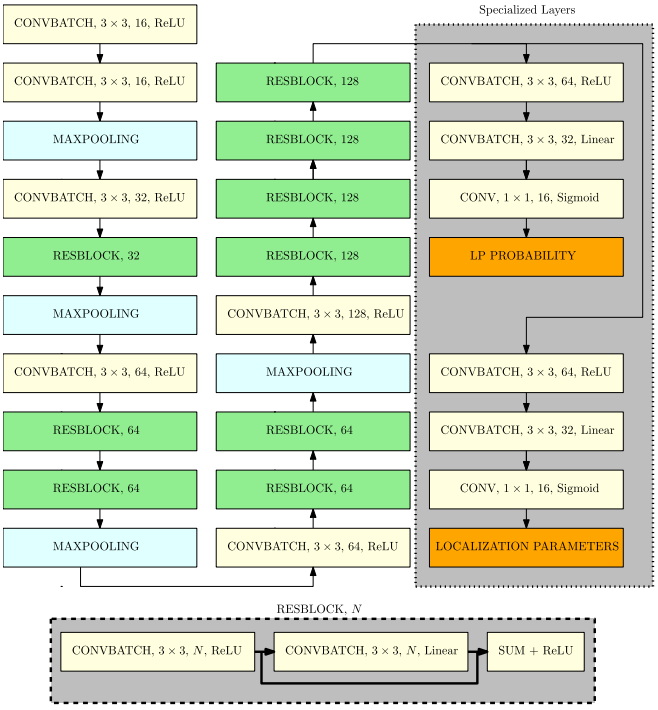

Fig. 1. Detailed IWPOD-NET architecture.

the very last layer (as in similar networks such as YOLO). However, as the variability of the appearance and shape of the input images increase, each task might back-propagate conflicting information to the previous layers, which might compromise one (or both) of the tasks. To alleviate this problem, we propose a simple solution by adding two shallow (but independent) sub-networks, one for each task.

The classification sub-network is formed by two *convbatch* layers, the last one with linear activation. A final layer with sigmoid activation provides the probability of having an LP at each cell of the output layer. The detection sub-network presents the same architecture, but the last layer regresses six parameters with a linear activation. The full description of the network (with the number of filters for each layer) is shown in Figure 1, and the parallel specialized sub-networks introduced in this work are highlighted.

2) The Loss Function: For an input image with height H and width W, and network stride given by $N_s = 2^4$, the output layer consists of an $M \times N \times 7$ volume, where $M = H/N_s$ and $N = W/N_s$. For each spatial point cell (m, n) in this layer, the network produces seven values: the first value $v_1(n, m)$ is the probability of encountering an object at the corresponding cell, and the following six values – $v_2(n, m)$ to $v_7(n, m)$ – are related to an affine transform used for LP localization. Hence, the loss function must involve the parameters related to both the classification and the localization tasks. We start by defining each individual loss, and then the final loss function used to train the model.

The first part of the loss function (classification) handles the probability $v_1(m, n)$ of having an object at a cell (m, n) of the network output. As in several binary classifiers, it is given by

$$f_{clas}(m, n) = \text{logloss}\left(\mathbb{I}(m, n), v_1(m, n)\right), \quad (1)$$

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

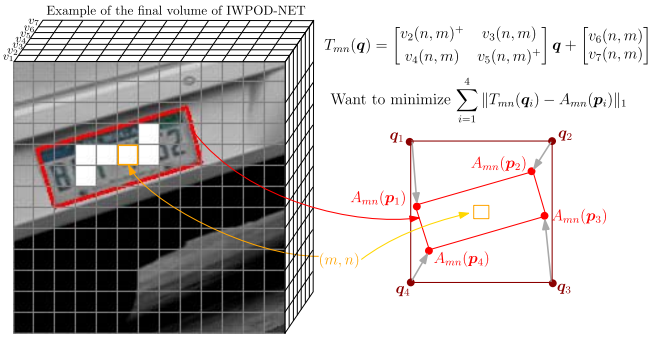

Fig. 2. Illustration of the affine transformation used to locate the LP in the final volume of IWPOD-NET.

where $\mathbb{I}(m,n)$ is the object indicator function that returns 1 if there is an object at cell (m,n) or 0 otherwise, and

$$\text{logloss}(y,\, p) = -y \log(p) - (1-y)\log(1-p). \tag{2}$$

To select the cells (m,n) in the output layer that should respond to an object, we first resize the corners $\boldsymbol{p}_i$ of the annotated plate to the output resolution (divide them by N_s). We then shrink the resized quadrilateral (with respect to the centroid) by a factor $\beta = 0.5$ and select all points (m,n) inside this region as positive labels for LP (i.e. $\mathbb{I}(m,n) = 1$). An illustration of such cells, shown in white, is given on the left of Figure 2 (original image from the Cars dataset [24]).

For the localization loss, the core idea is to find an affine transformation that maps a canonical square to the corners of the annotated LP. More precisely, let $\boldsymbol{p}_i = [x_i, y_i]^T$, for $i = 1, \cdots, 4$, denote the four corners of an annotated LP, which is given as input. They are ordered clockwise, starting from the top-left point. Also, let $\boldsymbol{q}_1 = [-0.5, -0.5]^T$, $\boldsymbol{q}_2 = [0.5, -0.5]^T$, $\boldsymbol{q}_3 = [0.5, 0.5]^T$, $\boldsymbol{q}_4 = [-0.5, 0.5]^T$ denote the corresponding vertices of a canonical unit square centered at the origin.

For a cell (m,n) in the output related to an LP (i.e., for which $\mathbb{I}(m,n) = 1$), we re-scale the points $\boldsymbol{p}_i$ by the inverse of the network stride, and re-center them according to (m,n), so that its center is roughly aligned with the origin. This is accomplished by applying a normalization function

$$A_{mn}(\boldsymbol{p}) = \frac{1}{\alpha}\left(\frac{1}{N_s}\boldsymbol{p} - \begin{bmatrix} n \\ m \end{bmatrix}\right), \tag{3}$$

where α is a scaling constant that aims to approximate the dimensions of the resized quadrilateral and the canonical square. We set $\alpha = 7.75$, which is the mean point between the maximum and minimum LP dimensions in the augmented training data divided by the network stride (training details will be provided next).

We then want to estimate the local affine transformation

$$T_{mn}(\boldsymbol{q}) = \begin{bmatrix} v_2(n,m)^+ & v_3(n,m) \\ v_4(n,m) & v_5(n,m)^+ \end{bmatrix} \boldsymbol{q} + \begin{bmatrix} v_6(n,m) \\ v_7(n,m) \end{bmatrix}, \tag{4}$$

such that $T_{mn}(\boldsymbol{q}_i) \approx A_{mn}(\boldsymbol{p}_i)$, for $i = 1, 2, 3, 4$, where $x^+ = \max\{0, x\}$ is the ReLU operator applied to the diagonal of the affine matrix, adopted to avoid horizontal or vertical flips. This process is illustrated in Figure 2. For the cell (m,n) related to the presence of an LP highlighted in orange on the left, we show on the right the scaled and centered quadrilateral $A_{mn}(\boldsymbol{p}_i)$, along with the canonical square and the expected mapping between the square and the quadrilateral (gray arrows). The quality of such mapping is quantified by the sum of the L^1 errors between the annotated and the warped points, given by

$$f_{loc}(m,n) = \sum_{i=1}^{4} \|T_{mn}(\boldsymbol{q}_i) - A_{mn}(\boldsymbol{p}_i)\|_1, \tag{5}$$

which is the localization loss.

The final loss is then obtained by adding the partial terms defined in Eqs. (1) and (5):

$$loss = \sum_{m=1}^{M}\sum_{n=1}^{N}[\mathbb{I}(m,n)\, f_{loc}(m,n) + \lambda f_{clas}(m,n)], \tag{6}$$

where λ is the relative weight of the localization term. Since the branching introduced in IWPOD-NET provides task-specific layers, each term in the loss is somewhat minimized independently, so that the choice for λ is not crucial (we used $\lambda = 1$). Also, note that the affine parameters are only estimated at cells related to an LP according to the GT annotation, i.e., when $\mathbb{I}(m,n) = 1$.

In the inference step, all spatial cells (n,m) in the final volume that present a classification probability $v_1(n,m)$ greater than a threshold T_{lp} are retrieved, and Non-Maxima Suppression (NMS) is applied to remove highly overlapping LPs. For the winning cells, the affine transformation T_{mn} is retrieved from the last six layers of the output volume and applied to the canonical points $\boldsymbol{q}_i$ according to Eq. (4). These points are finally de-normalized using the inverse operator of A_{nm} in Eq. (3) to retrieve the estimated LP corners in the original resolution. With the LP corners, we compute the planar homography that maps these points to the fronto-parallel appearance of the license plate. If the actual dimensions of the LP are known, we can use them to keep the correct aspect ratio of the plate. In this work, however, we unwarp the LP to a fixed-sized rectangle that relates to the network used to perform LPR, as it will be discussed in Section III-C.

3) Training Details: One advantage of using a fully convolutional network is that we can train it on a fixed resolution input (so that we can generate training batches and accelerate the process), but perform the inference on a test image with arbitrary size. Let us consider an input image with dimension $N \times N$, where N must be a multiple of the network stride $N_s = 16$.

Considering that the network must deal with a wide variety of LP templates, geometric variations (size and perspective distortion due to camera projection), as well as color and illumination conditions, we had to strongly rely on data augmentation to train the network. Given an annotated training image, we randomly apply geometric distortions (to emulate different sizes and viewpoints) and photometric distortions (to emulate color and illumination variations, as well as blur and noise). With this strategy, each training image can generate a limitless set of augmented versions. The augmentation process is described next.

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

We first define a target width $w \in [w_{min} w_{max}]$, with $0 < w_{min} < w_{max} < N$. We used $w_{min} = 0.2N$ and $w_{max} = N$, so that the largest width is five times the smallest. Since we do not have the actual dimensions of the LP (the training dataset comprises plates from several regions, each one with a different legislation about LP dimensions), we randomly select a target aspect ratio a_r for the plate and compute the height $h = w/a_r$. Based on the aspect ratios of vehicles and motorcycles in different regions, we select $a_r \in [2.5, 4.5]$ for vehicles, and $a_r \in [1.25, 2.5]$ for motorcycles. Then, we translate the target LP corners so that they fit entirely inside the desired dimension $N \times N$ by randomly selecting the horizontal offset $h_o \in [0, N - w]$ and the vertical offset $v_o \in [0, N - h]$ that translates the top-left coordinate of the target LP. In the next step, we find the planar homography H_0 that maps the annotated LP corners of the training image into the corners of the translated target LP. Note that applying H_0 to the input image would "rectify" the plate to a canonical fronto-parallel view.

Having the canonical plate, we can artificially apply a 3D rotation onto the plane that contains the LP, emulating a variety of viewpoints of the same camera. More precisely, we select the roll, pitch, and yaw angles within the values $\pm 45°$, $\pm 80°$, and $\pm 80°$, respectively. These ranges provide a very wide variability of viewpoints, and are used to generate another planar homography H_1 that performs the 3D rotation on a plane. The homography H_1 is obtained assuming a canonical pinhole camera with focal length $f = 1000$mm, and considering that the plane containing the LP lies at the same distance $z = 1000$mm. Finally, we compute the final homography $H = H_1 H_0$ and warp the input training image according to H (the annotation points are warped accordingly). Since the homographic warping typically does not fill the whole rectangular domain of the target image, a set of randomly selected background images is used for completion. This is done to increase the number of negative samples and hence reduce the number of false positives generated by the network.

After the geometrical distortions, photometric augmentation is applied. Each of the following distortions can be randomly applied:

- Negative: get the negative of the input image, with probability $p = 5\%$;
- Blur: apply Gaussian blur, with probability $p = 15\%$. The blur scale σ is randomly selected in the range $[0, 0.1N]$;
- Color change: random modifications in the HSV colorspace, with probability $p = 100\%$.

From the chosen set of transformations mentioned above, a great variety of augmented test images with very distinct visual characteristics can be obtained from a single manually labeled sample. Figure 3 illustrates the augmentation pipeline and the generation of four different augmented samples obtained from the same annotated input image, extracted from the Cars dataset.

To train the model, we used the 196 annotated images provided in [11] plus 175 images containing motorcycle LP annotations and 223 images with other vehicles (cars, buses, trucks), in a total of 693 annotated images. We chose $N = 208$ to define an input resolution of 208×208 to train the network, so all LPs in the augmentation process present widths varying from $w_{min} = 42$ to $w_{max} = 208$ pixels.

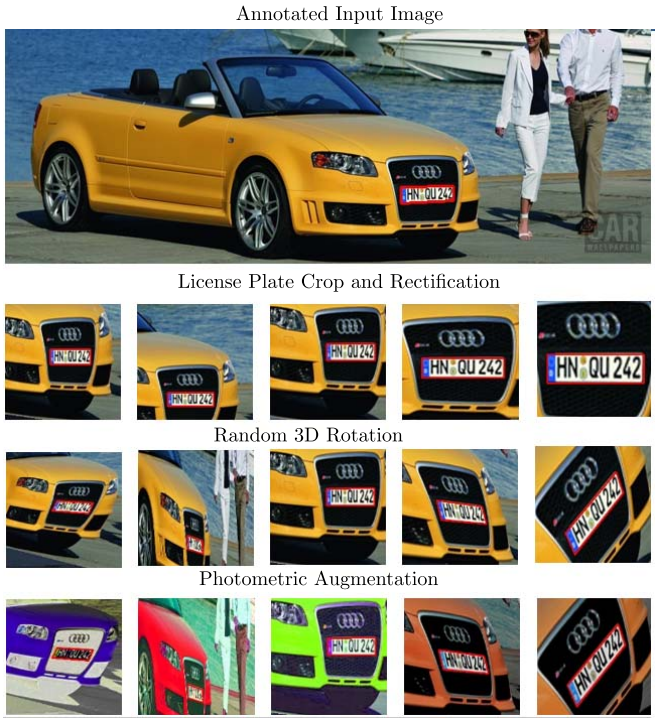

Fig. 3. Different augmentations for the same sample. The red quadrilateral represents the LP region throughout the augmentation process.

We copied the weights of first 18 layers of the original WPOD-NET (up to the penultimate residual layer) to the expanded model IWPOD-NET, froze them and trained the remaining layers for $40k$ iterations using mini-batches of size 64 and the ADAM optimizer [25] with a learning rate of 0.001. We then trained all layers using the same learning rate for more $50k$ iterations, reduced the learning rate to 0.0001, and trained for an additional $150k$ iterations.

B. Vehicle Detection and Resizing

As mentioned before, vehicle detection can be applied prior to running IWPOD-NET. Since IWPOD-NET was trained to detect LPs within a certain range, vehicle detection can be used both to reduce the search space for LPs and resize the image that is fed to IWPOD-NET, so that the expected LP size lies within the adequate range.

The SOTA on object detection is constantly advancing, and there are many detectors available. In this work, we opted to use Yolov3 [14], which provides a good compromise between accuracy and running times for object detection in general, but particularly in the context of vehicle detection [26]. Although the Yolov3 version trained on the COCO dataset already provides the desired classes – cars (including buses and trucks) and motorcycles –, training data involved scenarios that are not present (or desirable) in the context of ALPR. For example, vehicle annotations included small portions of a vehicle (e.g., only the hood as viewed by a dashboard-mounted camera), which might be interesting for a generic application involving vehicles, but not for ALPR. It also contained very

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

small vehicles, for which the license plates are not readable (or even barely visible). Besides pruning the annotations of the COCO 2017 and Pascal VOC 2012 datasets, we used additional images from the UFPR-ALPR datasets (training and validation sets) and private images containing motorcycles, ending up with 26,149 car and 6,661 motorcycle images. We changed the output layer of Yolov3 to accommodate only two classes, and transferred the weights of the Yolov3 body pre-trained on COCO for 10 epochs. We then unfroze all layers and trained for an additional 15 epochs with early stopping.

Since we want to handle a wide variety of capture scenarios, the dimensions of the input images and the relative sizes of the vehicles might vary considerably. Hence, we want to resize them to a suitable size before running IWPOD-NET. We note that in a mostly frontal or rear view, the variation of the fraction of the LP width w.r.t. the width of the vehicle crop is rather limited, since both structures have a limited size variation. In more oblique views, this fraction tends to be smaller, since the vehicle image also includes the lateral part. We also note that the aspect ratio of the vehicle bounding box provides a rough discrimination between frontal and oblique views: it is closer to one in frontal views, and increase as the viewpoint becomes more lateral.

Based on empirical evaluations of several datasets used in ALPR (they will be discussed in Section IV-B), we determined a scaling factor f_{sc} given by

$$f_{sc} = \min\left\{1, \frac{W_v a_r}{w_v}\right\}, \tag{7}$$

where $w_v \times h_v$ are the bounding box dimensions of a detected vehicle, $a_r = \max\{1, w_v/h_v\}$ is its (clipped) aspect ratio, and W_v is a scaling constant (set to 256 for cars and 208 for motorcycles).

C. License Plate Recognition (LPR)

Considering that IWPOD-NET produces a rectified LP, the LPR problem is highly simplified since it does not have to handle large character variations in terms of rotation or size. Still, there are several choices and possibilities for the LPR module. As discussed in Section II, some approaches explore sequential information (such as CTC) to directly obtain the LP string, while others rely on object detection for locating and classifying each character in the LP.

Despite the need of annotated character localization in the training step, we believe that LPR based on object detection is interesting for several reasons: i) since all character heights tend to be similar in the same LP, the character bounding boxes can be used to identify and reject outliers; ii) motorcycle LPs (or, in general, LPs that present multi-row characters) are a challenge to sequential/recurrent based methods, but can be handled when using detection-based methods; iii) if we know *a priori* the number of characters (or a range) in the LP, as regulated by several countries/regions, we can use the confidence scores to tune the number of detections accordingly. In this work, we present strategies for incorporating the above mention characteristics into any existing detection-based LPR method.

1) Imposing Size Consistency: Since the unwarping performed by IWPOD-NET provides an approximately frontal view of the LP, all characters in the LP should have approximately the similar height. We explore this knowledge by measuring the adherence of each character height (as provided by the bounding box) to the median value. If h_i is the height of the i^{th} detected character in a given LP, we compute the relative difference to the median as

$$e(h_i) = \left|\frac{h_i - \text{median}\{h_i\}}{\text{median}\{h_i\}}\right|, \tag{8}$$

and all values of $e(h_i)$ are expected to be low within the LP. Hence, characters with $e(h_i)$ larger than a threshold T_h are expected to be outliers.

To validate our hypothesis, we computed the median character height of each annotated LP in the dataset used to train the LPR network presented in [11], which comprises 5,429 characters with annotated bounding boxes distributed along 883 LPs. We then computed the relative median deviations $e(h_i)$ for all characters on the same plate. The mean value of all errors was 0.0267, with standard deviation of 0.0287, indicating a very homogeneous distribution with low values. We set a very conservative threshold of $T_h = 0.207$ that corresponds to the 99.9% percentile of the distribution (assuming normality), aiming to reduce the false removal of coherent characters.

2) Finding Two-Row Strings: In this work, we consider that motorcycle LPs present two rows of characters, as in Brazil and most European countries. When an LP is detected within a motorcycle vehicle label, we first retrieve the central vertical coordinate v_i of each character and wish to separate the detections in two rows. For that purpose, we compute the mean value of v_i in a robust manner (by eliminating the lowest and highest values, possibly related to outliers, and averaging the remaining characters). Then, characters presenting v_i smaller than the mean are assigned to the first row, and the others to the second row.

3) Imposing Region-Dependent Constraints: In the default region-agnostic mode, all detection results with score larger than a threshold T_{lpr} are retrieved. If we know that the LP must present a number $n \in [n_{min}, n_{max}]$ of characters, we can either relax the threshold T_{lpr} to obtain n_{min} detections (if $n < n_{min}$) or get only the characters with the highest n_{max} scores if $n > n_{max}$. Additionally to the per-character confidence threshold, we also impose a minimum "global confidence" to the whole LP, composed of the mean detection score of all characters. If this value is smaller than a threshold T_{mean}, the LP is discarded (this helps to eliminate false positives generated by LPD). It is also worth mentioned that a similar strategy (without the global confidence) was adopted in [15].

If the distribution of the characters in an LP is known (as in Brazil or China), we can also swap the occurrence of digits by similar letters (and vice-versa) when they are in the incorrect location. Examples of letters/digits pairs that can cause confusion are $B \leftrightarrow 8$, $Z \leftrightarrow 2$, $I \leftrightarrow 1$, $S \leftrightarrow 5$.

4) Integration of IWPOD-NET and LPR: As mentioned in Section III-A, the output of IWPOD-NET can be used to warp the detected LP to an arbitrary resolution. In this work,

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

we test two LPR networks: the OCR module presented in [11], which requires a 80×240 image as input, and the OCR used in [15], which was inspired on [11], trained with more data, and requires a 128×352 image. LPs related to cars are warped to these resolutions, whereas LPs related to motorcycles were warped to a smaller aspect ratio images (80×120 and 128×192 for each OCR network) and padded with a gray background, as in YOLO-based methods.

IV. Experimental Results

This section covers the experimental analysis of our full ALPR system. We first discuss existing validation datasets and evaluating metrics, then present the LPD results achieved by IWPOD-NET. Next, we provide the full ALPR results provided by our method, as well as comparisons with other SOTA methods and commercial systems.

A. Parameter Settings and Implementation Details

The proposed approach presents three networks in the pipeline, for which we empirically set the following acceptance thresholds: 0.35 for vehicle detection (YOLOv3), 0.35 for LP detection (IWPOD-NET), and 0.4 for character detection in LPR. Note that LP detection is only applied to regions produced by the vehicle detector, so we chose a lower threshold for the first module to reduce the number of false negatives. Also, LPs that either present less than four characters or that present an average character confidence smaller than $T_{mean} = 0.6$ (set empirically) are considered false positives and discarded.

In all experiments, results were produced using the default settings with no previous knowledge of the LP region. Results including any region-dependent post-processing step (as described in Section III-C3) were indicated explicitly. Also, since the characters "I" and "1" are identical for Brazilian LPs, they were considered as a single class in the evaluation of the OpenALPR-BR and UFPR-ALPR datasets.

The proposed CNN (IWPOD-NET) was implemented in the TensorFlow framework, with pre-trained weights available at https://github.com/claudiojung/iwpod-net. The YOLOv3-based vehicle detector was integrated using the Keras implementation available in https://github.com/qqwweee/keras-yolo3, and the YOLOv2-based LPR modules were integrated into our model using the Tensorflow implementation provided in https://github.com/thtrieu/darkflow. A Python wrapper was used to integrate all the modules. The hardware used for our experiments was an Intel i7 processor, with 16GB of RAM and an NVIDIA RTX 2070 GPU card with 8 GB RAM running on Windows 10 OS. The actual running times depend on the number of vehicles (if vehicle detection is used prior to IWPOD-NET) and the size of the input image fed to IWPOD-NET. We were able to run the full ALPR system with an average speed of 13.6 FPS (Frames per Second) considering all datasets. Our network with the trained weights will be made publicly available upon acceptance.

B. Evaluation Metrics and Datasets

Although the goal of ALPR systems is to correctly recognize all the LP characters, methods that present a full pipeline such as as [9], [20] present the results of both the LPD module and the final LPR result. For evaluating LPD, the typical protocol is to compute the Intersection over Union (IoU) between the detected LP and the GT annotation. Although widely adopted, we claim in this work that using the IoU might be misleading for two main reasons: i) LP annotations of most datasets are provided as rectangular bounding boxes, whereas the actual LP region is a quadrilateral, which might be considerably different in oblique/rotated views; ii) the IoU is not very informative about the final task: we can have two possible LPD results with the exact same IoU, but if one of them misses any character of the LP, the LPR task is compromised. Despite our criticism of the IoU, we present in Section IV-C a brief analysis of our LPD module using IWPOD-NET.

For the ultimate task of LPR (which is the final result of the proposed pipeline), we computed the total accuracy as defined in [1]. It is given as the number of correctly recognized LPs divided by the total number of annotated plates (which is actually the core idea of the recall rate). We also explore the well-known Levenshtein string distance [28], which is the minimum number of single-character edits required to transform one string into another. As a consequence, the distance is zero if and only if the strings match, which is equivalent to the accuracy. Our results for LPR are shown in Section IV-D.

As for the evaluation datasets, we first note that our method was designed and trained to deal with a variety of scenarios, with multiple vehicles (including cars, buses, trucks or bikes) presenting a variation of sizes and capture viewpoints, as well as license plates from different regions. As such, we chose four datasets available online, namely OpenALPR (EU and BR subsets),[1] UFPR-ALPR [8],[2] AOLP (subsets AC, LE and RP)[3] presented in [27], and CD-HARD presented in [11], which together cover a variety of viewing angles, illumination conditions, capture distances, and LP regions. It is also worth mentioning the CCPD dataset introduced in [20], which contains a very large set of Chinese LPs captured under a variety of situations with quadrilateral annotations for the LPs.[4] However, we downloaded the full dataset, and images were highly compressed, sometimes compromising even the readability of the LPs. In fact, the average file size was 71.8KB, whereas the authors mention an average file size of 200KB in the paper. For this reason, we used it only in the LPD experiments, described next.

C. License Plate Detection Results

For the LPD task, we use the AOLP and CCPD datasets. To cope with the particularities of each dataset, we employ a different evaluation for each one. Following [27] we use the bounding box annotations of AOLP and evaluate the results in terms of precision (PE) and recall (RE) rates, considering a detection correct if the IoU with a GT annotation is larger than 0.5. As in [11], vehicle detection was not used for the AOLP

[1] Available at https://github.com/openalpr/benchmarks.
[2] Available at https://web.inf.ufpr.br/vri/databases/ufpr-alpr/
[3] Available at https://github.com/HaoRecog/AOLP
[4] Available at https://github.com/detectRecog/CCPD

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

TABLE II
LPD PRECISION AND RECALL (IN %) FOR AOLP

Method	Subset					
	AC		LE		RP	
	PR	RE	PR	RE	PR	RE
Hsu et al. [27]	91	96	91	95	91	94
Li et al.[1]	98.5	98.4	97.8	97.6	95.3	95.6
Björklund et al. [9]	**100**	**99.3**	**99.8**	99.0	**99.8**	99.0
IWPOD-NET (Ours)	99.9	**99.3**	96.3	**99.7**	99.7	**100**

Fig. 4. Example of LPD for the AOLP dataset. Our results are shown in green, ground-truth annotations in red.

TABLE III
LPD ACCURACY (IN %) FOR CCPD

Method	Subset						
	DB	FN	Rot.	Tilt	Weath.	Chall.	Tot.
TE2E [3]	**91.7**	83.8	**95.1**	**94.5**	83.6	93.1	89.66
RPNet [20]	89.5	**85.3**	94.7	93.2	84.1	92.8	89.30
Ours	86.1	84.3	94.8	93.0	**95.7**	**93.4**	**89.71**
Ours (quad)	87.9	88.7	92.7	93.3	96.6	93.6	91.17

dataset, since the vehicles are already mostly framed (and running IWPOD-NET on the full image can provide a better measure of the precision). The results shown in Table II indicate that the proposed approach presents the best recall rates, but the precision is relatively lower. However, it is important to point out that IWPODNET can correctly locate some LPs that are not annotated in the GT data, which increases the number of false positives (and hence, the precision), particularly in the LE subset. If we disregard such detections, the precision for the subsets AC, LE and RP increases to 99.9%, 99.3%, and 99.8%. Figure 4 illustrates some results for AOLP, where our results are shown in green, and GT annotations in red. Examples of a "real" false positive, a "wrong" false positive and a false negative are shown in the second, third and fourth images, respectively, and are indicated by colored arrows.

For the CCPD dataset introduced in [20], the authors consider an LPD result correct if the IoU with the GT annotation is larger than 0.7, which is a more selective threshold. Although the dataset presents quadrilateral annotations for the LPs, the authors compute the IoU based on the bounding box, whereas we claim than comparing the IoU between the quadrilaterals is more adequate. Table III shows the results of our method and competitive approaches [3], [20] in different subsets of the CCPD dataset and the total accuracy (computed considering the number of samples in each subset). Although there is some variation across different subsets, the total accuracy of the three methods is very similar. We believe this is a remarkable result of our method, since the approaches [3], [20] were trained/fined-tuned using 100k images of the CCPD training set, whereas our method used only 105 images from this dataset (and 693 in total) for

Fig. 5. Example of LPD for the CCPD dataset. Our results are shown in green, ground-truth annotations in red.

TABLE IV
ABLATION STUDY FOR FOUR DATASETS

	OpenALPR		AOLP	CD-HARD
	EU	BR	RP	
WPOD-NET [11]	93.52%	91.23%	**98.36%**	75.00%
IWPOD-NET$^-$	94.44%	**94.74%**	**98.36%**	79.41%
IWPOD-NET	**97.22%**	**94.74%**	**98.36%**	**82.35%**

training our model. For the sake of illustration, We additionally show the accuracy comparing the IoU of the quadrilateral produced by our approach and the quadrilateral GT annotation, also using 0.7 as the acceptance threshold, shown as "Ours (quad)" in the table. We believe this is the correct metric to be used for evaluating LPD methods.

Figure 5 shows the results of our LPD module on a few selected samples of AOLP, and also on a random selection of the six subsets of the CCPD dataset. As can be observed, our method presents a strong overlap with the GT, and in some cases, it appears to capture the LP geometry even better than the GT annotations.

D. License Plate Recognition Results

We first present an ablation study comparing the method proposed in the original conference paper [11] and the proposed modifications, keeping the same OCR module for the LPR task. Table IV shows the final LPR accuracy of [11], the accuracy of our method changing the vehicle detector and the LPD module by IWPOD-NET (IWPOD-NET$^-$ in the table), and the final result adding the character post-processing given by Equation (8), called simply IWPOD-NET. We can note that just by changing the LPD module, we increase the accuracy in three out of four datasets (accuracy was the same for AOLP-RP), and by using the post-processing results yields

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

TABLE V
FINAL LPR RESULTS FOR OUR METHOD AND SOTA APPROACHS IN DIFFERENT DATASETS (IN %)

	OpenALPR		**UFPR-Test** (Framewise / Temporal)			**AOLP**				**CD-HARD**
	EU	**BR**	**Cars**	**Bikes**	**Total**	**AC**	**LE**	**RP**	**Total**	
IWPOD-NET	97.22	94.74	78.40 / 87.50	0.00 / 0.00	62.72 / 70.00	95.74	97.89	98.36	97.32	**82.35**
IWPOD-NET^{rg}	97.22	95.61	86.88 / 93.75	0.00 / 0.00	69.50 / 75.00	96.62	98.15	98.69	97.80	**82.35**
IWPOD-NET'	99.07	96.49	87.50 / 95.83	81.39 / **100.00**	86.28 / 96.67	96.04	98.02	98.20	97.41	68.63
IWPOD-NET'^{rg}	**99.07**	**98.25**	93.75 / 97.92	**83.89 / 100.00**	**91.78 / 98.83**	**98.38**	**99.21**	**99.51**	99.02	68.63
	Literature									
Silva & Jung [11]	93.52	91.23	-	-	-	93.10	95.51	98.36	94.82	75.00
Laroca et al. [8]	-	-	72.2 / 83.3	35.6 / 58.3	64.9 / 78.3	-	-	-	-	-
Laroca et al. [15]	96.9	-	**95.9 / 98.3**	66.3 / 70.0	90.0 / 92.7	-	-	-	**99.2***	-
Li et al. [1]	-	-	-	-	-	94.85	94.19	88.38	92.68	-
Li et al. [3]	-	-	-	-	-	95.29	96.57	83.63	92.35	-
Hsu et al. [29]	-	-	-	-	-	-	-	85.70^{+}	-	-
Björklund et al. [9]	-	-	-	-	-	94.60	97.80	96.90	96.47	-
Selmi et al. [10]	-	-	-	-	-	97.80	97.40	96.30	97.20	-
	Commercial systems									
OpenALPR	88.89	97.37	58.0 / 89.6	22.8 / 0.0	50.9 / 71.7	92.22	91.09	93.62	91.78	59.62
Sighthound	94.44	**98.25**	58.4 / 70.8	3.30 / 0.0	47.40 / 56.7	94.71	95.61	94.93	95.10	72.12

*Value computed for a test subset with 683 images from all subsets (remaining images used to train), as informed in [15].
$^{+}$In [29] the authors provided an estimate, and not the real evaluation.

additional gain. As we will show next, results can get even better by changing the LPR module.

Table V shows the final ALPR accuracy results of our method, SOTA approaches and commercial systems. About the commercial systems, we ran the results for OpenALPR and Sighthound in July 2020 for all datasets, except for the UFPR dataset that already presents the benchmarks for these two systems. For OpenALPR, the LP region must be informed (CD-HARD is multi-region, we informed the US since it seems to be the most flexible region), whereas Sighthound is region-agnostic. Regarding our method, results with ' were obtained with the OCR of [15], and superscript ^{rg} indicates the additional use of regional information as described in Section III-C3. For Brazilian LPs, that means imposing seven characters and keeping the LLLDDDD structure of the plate (L is a letter, D is a digit); for Taiwanese LPs, we enforce five or six characters; for European LPs, the variability is large and adding regional information does not change the results.

We can observe that the region-agnostic version of IWPOD-NET presents very competitive results, being the top-scorer for CD-HARD and one of the top ones for all AOLP-subsets. It is worth mentioning that several approaches, such as [1], [3], [8], [9], [27], present methods tailored to a specific region or dataset, whereas the exact same version of IWPOD-NET was used in all tests. When including region-dependent information, the results get even better for Brazilian and Taiwanese LPs. Our best results for most datasets were achieved when using a better trained OCR method (IWPOD-NET' and IWPOD-NET'^{rg}), achieving SOTA accuracy results in most subsets.

It is important to note that the UFPR-ALPR dataset presents several images of the same vehicle/LP (the test set contains 30 frames for 60 different vehicles), so that temporal coherence can be explored. If we employ the same weighted mode voting scheme proposed in [19], our temporally-consistent results improve considerably when compared to the frame-wise version. Also, note that the accuracy for IWPOD-NET and IWPOD-NET^{rg} is zero because the OCR used in these versions of our method was not trained with motorcycle LPs. When changing the OCR module (IWPOD-NET' and IWPOD-NET'^{rg}), our results yielded the best result among all methods, achieving 100% accuracy for motorcycles when adding temporal consistency.

Figure 6 shows a few examples of ALPR using the proposed approach (version IWPOD-NET') applied to different datasets. These images were extracted from (ordering from top-left) the Cars dataset (images 1, 2, 6), AOLP (images 3, 8-10), OpenALPR (images 7, 11) and UFPR (images 4, 5). The green rectangles are the outputs of the vehicle detector, the red quadrilateral indicates a detected LP, and a yellow quadrilateral indicates an LPD result that was discarded (either because the mean OCR confidence was too low, or the number of detected characters was smaller than four). The rectified LP and the corresponding LPR result are shown on the top or bottom of the images. The top row of Figure 6 shows correct LPR results, and the first two images present very distorted LPs that were correctly rectified by IWPOD-NET. The middle image shows a correct LPR and also a partial LP detection of another vehicle. The last two images show results for a bus and motorcycle, respectively. The bottom row of Figure 6 shows some failure cases. In the first image, only three of the four characters were recognized, and the LP was mistakenly discarded. The second and third images show examples were IWPOD-NET could not identify the LP region correctly, which led to a failure in LPR. In the fourth image, the LP was not detected, and in the fifth image, a phone number on the back of a vehicle (top right) was wrongly detected as an LP (false positive result). Finally, the last image illustrates the most common failure cause: the LP was correctly detected and rectified, and the error occurs in the LPR module.

Finally, we show in Table VI the results of the proposed method (different versions) also considering 1- and 2-character

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

Fig. 6. Examples of final ALPR obtained with the proposed method.

TABLE VI
ACCURACY OF OUR METHOD FOR ALL DATASETS (IN %) CONSIDERING A TOLERANCE

	OpenALPR		**UFPR-Test** (Framewise / Temporal)			**AOLP**			**CD-HARD**
	EU	**BR**	**Cars**	**Bikes**	**Total**	**AC**	**LE**	**RP**	
	1-character tolerance								
IWPOD-NET	99.07	97.37	93.96 / 100.00	0.00 / 0.00	75.17 / 80.00	98.83	99.74	100.00	97.06
IWPOD-NET^{rg}	99.07	98.25	96.25 / 80.00	0.00 / 0.00	77.00 / 80.00	99.12	99.74	100.00	97.06
IWPOD-NET'	99.07	98.25	97.71 / 100.00	91.11 / 100.00	96.39 / 100.00	99.27	99.74	100.00	95.10
IWPOD-NET'^{rg}	99.07	98.25	97.85 / 100.00	91.39 / 100.00	96.56 / 100.00	99.27	99.74	100.00	95.10
	2-character tolerance								
IWPOD-NET	100.00	100.00	97.22 / 100.00	0.56 / 0.00	75.89 / 80.00	99.27	99.74	100.00	98.04
IWPOD-NET^{rg}	100.00	98.25	97.50 / 100.00	0.56 / 0.00	78.11 / 80.00	99.27	99.74	100.00	98.04
IWPOD-NET'	100.00	99.12	98.06 / 100.00	92.22 / 100.00	96.89 / 100.00	99.27	99.74	100.00	98.04
IWPOD-NET'^{rg}	100.00	99.12	97.99 / 100.00	92.22 / 100.00	96.83 / 100.00	99.27	99.74	100.00	98.04

tolerances in the string edit distance. For some datasets, we achieve 100% accuracy, which is also an indirect clue that the LPD module can correctly locate the LP.

V. CONCLUSION AND FUTURE WORK

In this work, we presented a complete ALPR system for unconstrained scenarios, where the LP might be considerably distorted due to oblique views. The main contribution of the work was the introduction of a network capable of detecting and unwarping multiple distorted LPs by regressing an affine transformation that maps a canonical square to the corners of the LP. This network was trained with a small amount of annotated data (only 693 images), and showed remarkable generalization to handle both car and motorcycle LPs captured at a variety of illumination conditions and viewpoints. Additional contributions were the introduction of post-processing strategies than can be coupled to any detector-based license plate recognition module, and the use of a fine-tuned an object detector to identify cars (including buses and trucks) and motorcycles as output classes.

Our results indicate that the proposed LPD approach presents accuracy comparable to other SOTA methods training with much more data than ours in a challenging dataset (CCPD). When combined to object-based LPR modules, our full ALPR method outperforms existing methods in challenging datasets containing LPs captured at strongly oblique views while still maintaining SOTA accuracy results for more controlled capture scenarios. Regarding the specific problem of motorcycle LPR, the proposed approach outperformed existing methods [8], [15], achieving zero error rate when temporal information is also explored.

For future work, we plan to concentrate on the recognition module to handle highly degraded LPs (e.g., low-resolution or motion blurred). For this task, we intend to strongly rely on temporal information to improve the quality of the LP by merging deep features from several frames before the recognition step, unlike the temporal consistency method explored in this work (which performs LPR at each frame and then applies consistency to the LPR results).

REFERENCES

[1] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMs," Jan. 2016, *arXiv:1601.05610*. [Online]. Available: http://arxiv.org/abs/1601.05610

[2] S. Montazzolli and C. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks," in *Proc. 30th Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2017, pp. 55–62.

[3] 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. 2018.

[4] F. D. Kurpiel, R. Minetto, and B. T. Nassu, "Convolutional neural networks for license plate detection in images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3395–3399. [Online]. Available: http://ieeexplore.ieee.org/document/8296912/

[5] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2351–2363, Sep. 2017.

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.

[6] G.-S. Hsu, A. Ambikapathi, S.-L. Chung, and C.-P. Su, "Robust license plate detection in the wild," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/8078493/

[7] 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.

[8] R. Laroca *et al.*, "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.

[9] 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.

[10] Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, "DELP-DAR system for license plate detection and recognition," *Pattern Recognit. Lett.*, vol. 129, pp. 213–223, Jan. 2020.

[11] 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.

[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. [Online]. Available: http://ieeexplore.ieee.org/document/7780460/

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525. [Online]. Available: http://arxiv.org/abs/1612.08242 and http://ieeexplore.ieee.org/document/8100173/

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[15] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," 2019, *arXiv:1909.01754*. [Online]. Available: http://arxiv.org/abs/1909.01754

[16] 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/

[17] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[18] 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.

[19] 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.

[20] 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.

[21] J. Spanhel, J. Sochor, R. Juranek, A. Herout, L. Marsik, and P. Zemcik, "Holistic recognition of low quality license plates by CNN using track annotated data," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[22] Y. Lee, J. Lee, H. Ahn, and M. Jeon, "SNIDER: Single noisy image denoising and rectification for improving license plate recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1017–1026.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, vol. 4, no. 3, pp. 770–778.

[24] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sydney, Australia, Dec. 2013, pp. 554–561.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–15, Dec. 2014.

[26] D. Impedovo, F. Balducci, V. Dentamaro, and G. Pirlo, "Vehicular traffic congestion classification by visual features and deep learning approaches: A comparison," *Sensors*, vol. 19, no. 23, p. 5213, Nov. 2019.

[27] 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.

[28] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet Physics Doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

Sergio M. Silva received the Ph.D. degree in computer vision from the Federal University of Rio Grande do Sul, Brazil, in 2019. He is currently an Adjunct Professor with the State University of Londrina, and is also working as an independent AI Researcher and Developer. His main research interests are machine learning and computer vision.

Cláudio Rosito Jung (Senior Member, IEEE) received the B.S. and M.S. degrees in applied mathematics and the Ph.D. degree in computer sciences from the Universidade Federal do Rio Grande do Sul (UFRGS), Brazil, in 1993, 1995, and 2002, respectively. He is currently a Faculty Member at the Computer Science Department, UFRGS, and was a Visiting Faculty at the University of Pennsylvania from July 2015 to July 2016. His research interests include several aspects in image processing, computer vision, and pattern recognition, such as medical imaging, multiscale image analysis, intelligent vehicles, multimedia applications, human motion synthesis and analysis, audiovisual signal processing, and stereo/multiview matching.

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:43:13 UTC from IEEE Xplore. Restrictions apply.