License Plate Detection and Recognition in Unconstrained Scenarios

Sérgio Montazzolli Silva[0000−0003−2444−3175] and Cláudio Rosito Jung[0000−0002−4711−5783]

Institute of Informatics - Federal University of Rio Grande do Sul
Porto Alegre, Brazil
{smsilva,crjung}@inf.ufrgs.br

Abstract. Despite the large number of both commercial and academic methods for Automatic License Plate Recognition (ALPR), most existing approaches are focused on a specific license plate (LP) region (e.g. European, US, Brazilian, Taiwanese, etc.), and frequently explore datasets containing approximately frontal images. This work proposes a complete ALPR system focusing on unconstrained capture scenarios, where the LP might be considerably distorted due to oblique views. Our main contribution is the introduction of a novel Convolutional Neural Network (CNN) capable of detecting and rectifying multiple distorted license plates in a single image, which are fed to an Optical Character Recognition (OCR) method to obtain the final result. As an additional contribution, we also present manual annotations for a challenging set of LP images from different regions and acquisition conditions. Our experimental results indicate that the proposed method, without any parameter adaptation or fine tuning for a specific scenario, performs similarly to state-of-the-art commercial systems in traditional scenarios, and outperforms both academic and commercial approaches in challenging ones.

Keywords: License Plate · Deep learning · Convolutional Neural Networks

1 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 improve 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 [18,28,19,3,2,9,31,17]. Along with academic papers, several commercial ALPR systems have been also exploring 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/), the commercial version of OpenALPR (http://www.openalpr.com/) and Amazon Rekognition (https://aws.amazon.com/rekognition/).

Fig. 1: Examples of challenging oblique license plates present in the proposed evaluation dataset.

Despite the advances in the state-of-the-art, most ALPR systems assume a mostly frontal view of the vehicle and LP, which is common in applications such as toll monitoring and parking lot validation, for instance. However, more relaxed image acquisition scenarios (e.g. a law enforcement agent walking with a mobile camera or smartphone) might lead to oblique views in which the LP might be highly distorted yet still readable, as illustrated in Fig. 1, and for which even state-of-the-art commercial systems struggle.

In this work we propose a complete ALPR system that performs well over a variety of scenarios and camera setups. Our main contribution is the introduction of a novel network capable of detecting the LP in many different camera poses and estimate its distortion, allowing a rectification process before OCR. An additional contribution is the massive use of synthetically warped versions of real images for augmenting the training dataset, allowing the network to be trained from scratch using less than 200 manually labeled images. The proposed network and data augmentation scheme also led to a flexible ALPR system that was able to successfully detect and recognize LPs in independent test datasets using the same system parametrization.

We also generalized an existing OCR approach develpoed for Brazilian LPs [28]. Basically, we re-trained their OCR network using a new training set composed by a mixture of real and artificially generated data using font-types similar to the target regions. As a result, the re-trained network became much more robust for detection and classification of real characters in the original Brazilian scenario, but also for European and Taiwanese LPs, achieving very high precision and recall rates. All the annotated data used for this work is publicly available[1], and the reference images can be obtained by downloading the Cars Dataset [16], the SSIG Database [6], and the AOLP dataset [10].

[1] Available at http://www.inf.ufrgs.br/~crjung/alpr-datasets.

The remainder of this work is organized as follows. In Section 2 we briefly review related approaches toward ALPR. Details of the proposed method are given in Section 3, where we describe the LP detection and unwarping network, as well as the data augmentation process used to train our models. The overall evaluation and final results are presented in Section 4. Finally, Section 5 summarizes our conclusions and gives perspectives for some future work.

2 Related Work

ALPR is the task of finding and recognizing license plates in images. It is commonly broken into four subtasks that form a sequential pipeline: vehicle detection, license plate detection, character segmentation and character recognition. For simplicity, we refer to the combination of the last two subtasks as OCR.

Many different ALPR systems or related subtasks have been proposed in the past, typically using image binarization or gray-scale analysis to find candidate proposals (e.g. LPs and characters), followed by handcrafted feature extraction methods and classical machine learning classifiers [1,4]. With the rise of DL, the state-of-the-art started moving to another direction, and nowadays many works employ CNNs due to its high accuracy for generic object detection and recognition [23,24,21,25,8,11].

Related to ALPR are Scene Text Spotting (STS) and number reading in the wild (e.g. from Google Street View images [22]) problems, which goals are to find and read text/numbers in natural scenes. Although ALPR could be seen as a particular case of STS, the two problems present particular characteristics: in ALPR, we need to learn characters and numbers (without much font variability) with no semantic information, while STS is focused on textual information containing high font variability, and possibly exploring lexical and semantic information, as in [30]. Number reading does not present semantic information, but dealing only with digits is simpler than the ALPR context, since it avoids common digit/letter confusions such as B-8, D-0, 1-I, 5-S, for instance.

As the main contribution of this work is a novel LP detection network, we start this section by reviewing DL-based approaches for this specific subtask, as well as a few STS methods that can handle distorted text and could be used for LP detection. Next, we move to complete ALPR DL-based systems.

2.1 License Plate Detection

The success of YOLO networks [23,24] inspired many recent works, targeting real-time performance for LP detection [28,9,31,17]. A slightly modified version of the YOLO [23] and YOLOv2 [24] networks were used by Hsu et al. [9], 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 over the bounding boxes extracted. 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 is needed.

In [31], a setup of two YOLO-based networks was trained with the goal of detecting 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, and not more complex deformations caused by oblique camera views, such as the ones illustrated in Fig. 1. Also, as they do not present a complete ALPR system, it is difficult to evaluate how well an OCR method would perform on the detected regions.

License plate detectors using sliding window approaches or candidate filtering coupled with CNNs can also be found in the literature [3,2,27]. 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 [21] and Faster R-CNN [25].

Although Scene Text Spotting (STS) methods focus mostly on large font variations and lexical/semantic information, but it is worth mentioning a few approaches that deal with rotated/distorted text and could be explored for LP detection in oblique views. Jaderberg and colleagues [13] presented a CNN-based approach for text recognition in natural scenes using an entirely synthetic dataset to train the model. Despite the good results, they strongly rely on N-grams, which are not applicable to ALPR. Gupta et al. [7] also explored synthetic dataset by realistically pasting text into real images, focusing mostly on text localization. The output is a rotated bounding box with around the text, which finds limitations for off-plane rotations common in ALPR scenarios.

More recently, Wang et al. [29] presented an approach to detect text in a variety of geometric positions, called Instance Transformation Network (ITN). It is basically a composition of three CNNs: a backbone network to compute features, a transformation network to infer affine parameters where supposedly exists text in the feature map, and a final classification network whose input is built by sampling features according to the affine parameters. Although this approach can (in theory) handle off-plane rotations, it is not able to correctly infer the transformation that actually maps the text region to a rectangle, since there is no physical (or clear psychological) bounding region around the text that should map to a rectangle in an undistorted view. In ALPR, the LP is rectangular and planar by construction, and we explore this information to regress the transformation parameters, as detailed in Section 3.2.

2.2 Complete ALPR Methods

The works of Silva and Jung [28] and Laroca et al. [17] presented complete ALPR systems based on a series of modified YOLO networks. Two distinct networks were used in [28], one to jointly detect cars and LPs, and another to perform OCR. A total of five networks were used in [17], basically one for each ALPR subtask, being two for character recognition. Both reported real-time systems,

but they are focused only on Brazilian license plates and were not trained to capture distortion, only frontal and nearly rectangular LPs.

Selmi et al. [27] 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.

Li et al. [19] presented a network based on Faster R-CNN [25]. Shortly, a Region Proposal Network is assigned to find candidate LP regions, whose corresponding feature maps are cropped by a RoI Pooling layer. Then, these candidates are fed into the final part of the network, which computes the probability of being/not being an LP, and performs OCR through a Recurrent Neural Network. Despite promising, the evaluation presented by the authors shows a lack of performance in most challenging scenarios containing oblique LPs.

Commercial systems are good reference points to the state-of-the-art. 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. As mentioned in Section 1, examples are Sighthound, OpenALPR (which is an official NVIDIA partner in the Metropolis platform[2]) and Amazon Rekognition (a general-purpose AI engine including a text detection and recognition module that can be used for LP recognition, as informed by the company).

3 The Proposed Method

The proposed approach is composed by three main steps: vehicle detection, LP detection and OCR, as illustrated in Fig. 2. Given an input image, the first module detects vehicles in the scene. Within each detection region, the proposed Warped Planar Object Detection Network (WPOD-NET) searches for LPs and regresses one affine transformation per detection, allowing a rectification of the LP area to a rectangle resembling a frontal view. These positive and rectified detections are fed to an OCR Network for final character recognition.

3.1 Vehicle Detection

Since vehicles are one of the underlying objects present in many classical detection and recognition datasets, such as PASCAL-VOC [5], ImageNet [26], and COCO [20], we decided to not train a detector from scratch, and instead chose a known model to perform vehicle detection considering a few criteria. On one hand, a high recall rate is desired, since any miss detected vehicle having a visible LP leads directly to an overall LP miss detection. On the other hand, high

[2] NVIDIA platform for video analysis in smart cities (https://www.nvidia.com/en-us/autonomous-machines/intelligent-video-analytics-platform/).

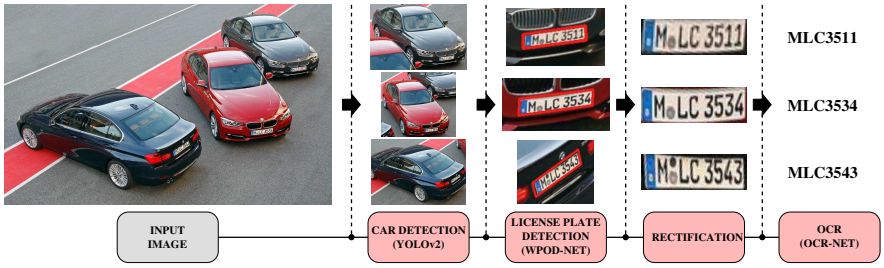

Fig. 2: Illustration of the proposed pipeline.

precision is also desirable to keep running times low, as each falsely detected vehicle must be verified by WPOD-NET. Based on these considerations, we decided to use the YOLOv2 network due to its fast execution (around 70 FPS) and good precision and recall compromise (76.8% mAP over the PASCAL-VOC dataset). We did not perform any change or refinement to YOLOv2, just used the network as a black box, merging the outputs related to vehicles (i.e. cars and buses), and ignoring the other classes.

The positive detections are then resized before being fed to WPOD-NET. As a rule of thumb, larger input images allow the detection of smaller objects but increase the computational cost [12]. In roughly frontal/rear views, the ratio between the LP size and the vehicle bounding box (BB) is high. However, this ratio tends to be much smaller for oblique/lateral views, since the vehicle BB tends to be larger and more elongated. Hence, oblique views should be resized to a larger dimension than frontal ones to keep the LP region still recognizable.

Although 3D pose estimation methods such as [32] might be used to determine the resize scale, this work presents a simple and fast procedure based on the aspect ratio of the vehicle BB. When it is close to one, a smaller dimension can be used, and it must be increased as the aspect ratio gets larger. More precisely, the resizing factor f_{sc} is given by

$$f_{sc} = \frac{1}{\min\{W_v, H_v\}} \min\left\{ D_{min} \frac{\max(W_v, H_v)}{\min(W_v, H_v)}, D_{max} \right\}, \tag{1}$$

where W_v and H_v are the width and height of the vehicle BB, respectively. Note that $D_{min} \leq f_{sc} \min(W_v, H_v) \leq D_{max}$, so that D_{min} and D_{max} delimit the range for the smallest dimension of the resized BB. Based on experiments and trying to keep a good compromise between accuracy and running times, we selected $D_{min} = 288$ and $D_{max} = 608$.

3.2 License Plate Detection and Unwarping

License plates are intrinsically rectangular and planar objects, which are attached to vehicles for identification purposes. To take advantage of its shape, we proposed a novel CNN called Warped Planar Object Detection Network. This

network learns to detect LPs in a variety of different distortions, and regresses coefficients of an affine transformation that "unwarps" the distorted LP into a rectangular shape resembling a frontal view. Although a planar perspective projection could be learned instead of the affine transform, the division involved in the perspective transformation might generate small values in the denominator, and hence leading to numerical instabilities.

The WPOD-NET was developed using insights from YOLO, SSD and Spatial Transformer Networks (STN) [14]. YOLO and SSD perform fast multiple object detection and recognition at once, but they do not take spatial transformations into account, generating only rectangular bounding boxes for every detection. On the opposite, STN can be used for detecting non-rectangular regions, however it cannot handle multiple transformations at the same time, performing only a single spatial transformation over the entire input.

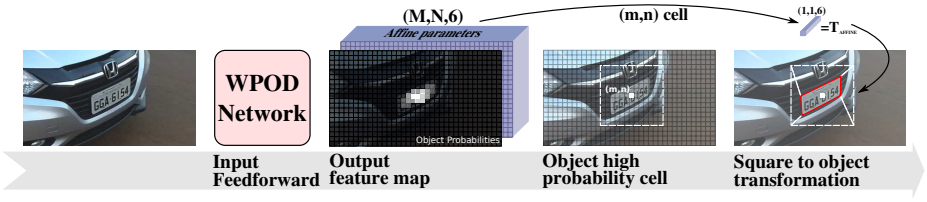

Fig. 3: Fully convolutional detection of planar objects (cropped for better visualization).

The detection process using WPOD-NET is illustrated in Fig. 3. Initially, the network is fed by the resized output of the vehicle detection module. The feedforwarding results in an 8-channel feature map that encodes object/non-object probabilities and affine transformation parameters. To extract the warped LP, let us first consider an imaginary square of fixed size around the center of a cell (m, n). If the object probability for this cell is above a given detection threshold, part of the regressed parameters is used to build an affine matrix that transforms the fictional square into an LP region. Thus, we can easily unwarp the LP into a horizontally and vertically aligned object.

Network Architecture The proposed architecture has a total of 21 convolutional layers, where 14 are inside residual blocks [8]. The size of all convolutional filters is fixed in 3×3. ReLU activations are used throughout the entire network, except in the detection block. There are 4 max pooling layers of size 2×2 and stride 2 that reduces the input dimensionality by a factor of 16. Finally, the detection block has two parallel convolutional layers: (i) one for inferring the probability, activated by a softmax function, and (ii) another for regressing the affine parameters, without activation (or, equivalently, using the identity $F(\boldsymbol{x}) = \boldsymbol{x}$ as the activation function).

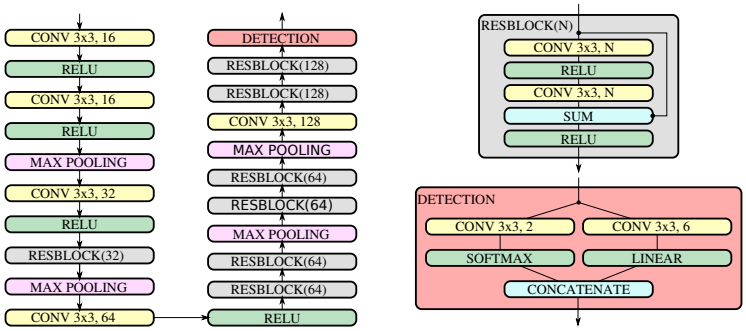

Fig. 4: Detailed WPOD-NET architecture.

Loss Function Let $\boldsymbol{p}_i = [x_i, y_i]^T$, for $i = 1, \cdots, 4$, denote the four corners of an annotated LP, clockwise starting from top-left. 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 an input image with height H and width W, and network stride given by $N_s = 2^4$ (four max pooling layers), the network output feature map consists of an $M \times N \times 8$ volume, where $M = H/N_s$ and $N = W/N_s$. For each point cell (m, n) in the feature map, there are eight values to be estimated: the first two values (v_1 and v_2) are the object/non-object probabilities, and the last six values (v_3 to v_8) are used to build the local affine transformation T_{mn} given by:

$$T_{mn}(\boldsymbol{q}) = \begin{bmatrix} \max(v_3, 0) & v_4 \\ v_5 & \max(v_6, 0) \end{bmatrix} \boldsymbol{q} + \begin{bmatrix} v_7 \\ v_8 \end{bmatrix}, \tag{2}$$

where the max function used for v_3 and v_6 was adopted to ensure that the diagonal is positive (avoiding undesired mirroring or excessive rotations).

To match the network output resolution, the points $\boldsymbol{p}_i$ are re-scaled by the inverse of the network stride, and re-centered according to each point (m, n) in the feature map. 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 represents the side of the fictional 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.

Assuming that there is an object (LP) at cell (m, n), the first part of the loss function considers the error between a warped version of the canonical square and the normalized annotated points of the LP, given by

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

The second part of the loss function handles the probability of having/not having an object at (m, n). It is similar to the SSD confidence loss [21], and basically is the sum of two log-loss functions

$$f_{probs}(m,n) = \text{logloss}(\mathbb{I}_{obj}, v_1) + \text{logloss}(1 - \mathbb{I}_{obj}, v_2), \tag{5}$$

where $\mathbb{I}_{obj}$ is the object indicator function that returns 1 if there is an object at point (m, n) or 0 otherwise, and $\text{logloss}(y, p) = -y \log(p)$. An object is considered inside a point (m, n) if its rectangular bounding box presents an IoU larger than a threshold γ_{obj} (set empirically to 0.3) w.r.t. another bounding box of the same size and centered at (m, n).

The final loss function is given by a combination of the terms defined in Eqs. (4) and (5):

$$loss = \sum_{m=1}^{M} \sum_{n=1}^{N} [\mathbb{I}_{obj} f_{affine}(m,n) + f_{probs}(m,n)]. \tag{6}$$

Training Details For training the proposed WPOD-NET, we created a dataset with 196 images, being 105 from the Cars Dataset, 40 from the SSIG Dataset (training subset), and 51 from the AOLP dataset (LE subset). For each image, we manually annotated the 4 corners of the LP in the picture (sometimes more than one). The selected images from the Cars Dataset include mostly European LPs, but there are many from the USA as well as other LP types. Images from SSIG and AOLP contain Brazilian and Taiwanese LPs, respectively. A few annotated samples are shown in Fig. 5.

Fig. 5: Examples of the annotated LPs in the training dataset.

Given the reduced number of annotated images in the training dataset, the use of data augmentation is crucial. The following augmentation transforms are used:

- Rectification: the entire image is rectified based on the LP annotation, assuming that the LP lies on a plane;
- Aspect-ratio: the LP aspect-ratio is randomly set in the interval $[2, 4]$ to accommodate sizes from different regions;

– Centering: the LP center becomes the image center;
– Scaling: the LP is scaled so its width matches a value between $40px$ and $208px$ (set experimentally based on the readability of the LPs). This range is used to define the value of α used in Eq. (3);
– Rotation: a 3D rotation with randomly chosen angles is performed, to account for a wide range of camera setups;
– Mirroring: 50% chance;
– Translation: random translation to move the LP from the center of the image, limited to a square of 208×208 pixels around the center;
– Cropping: considering the LP center before the translation, we crop a 208×208 region around it;
– Colorspace: slight modifications in the HSV colorspace;
– Annotation: the locations of the four LP corners are adjusted by applying the same spatial transformations used to augment the input image.

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. For example, Fig. 6 shows 20 different augmentation samples obtained from the same image.

Fig. 6: Different augmentations for the same sample. The red quadrilateral represents the transformed LP annotation.

We trained the network with $100k$ iterations of mini-batches of size 32 using the ADAM optimizer [15]. The learning rate was set to 0.001 with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The mini-batches were generated by randomly choosing and augmenting samples from the training set, resulting in new input tensors of size $32 \times 208 \times 208 \times 3$ at every iteration.

3.3 OCR

The character segmentation and recognition over the rectified LP is performed using a modified YOLO network, with the same architecture presented in [28]. However, the training dataset was considerably enlarged in this work by using synthetic and augmented data to cope with LP characteristics of different regions around the world (Europe, United States and Brazil)[3].

[3] We also used Taiwanese LPs, but could not find information in English about the font type used by this country in order to include in the artificial data generation.

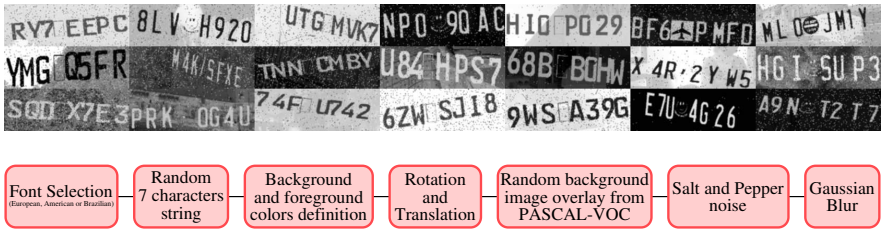

Fig. 7: Artificial LP samples with the proposed generation pipeline (bottom).

The artificially created data consist of pasting a string of seven characters onto a textured background and then performing random transformations, such as rotation, translation, noise, and blur. Some generated samples and a short overview of the pipeline for synthetic data generation are shown in Fig. 7. As shown in Section 4, the use of synthetic data helped to greatly improve the network generalization, so that the exact same network performs well for LPs of different regions around the world.

3.4 Evaluation Datasets

One of our goals is to develop a technique that performs well in a variety of unconstrained scenarios, but that should also work well in controlled ones (such as mostly frontal views). Therefore, we chose four datasets available online, namely OpenALPR (BR and EU)[4], SSIG and AOLP (RP), which cover many different situations, as summarized in the first part of Table 1. We consider three distinct variables: LP angle (frontal and oblique), distance from vehicles to the camera (close, intermediate and far), and the region where the pictures were taken.

Table 1: Evaluation datasets.

Database (subset)	LP angle	Vehicle Dist.	#images	Region
OpenALPR [5] (EU)	mostly frontal	close	104	Europe
OpenALPR (BR)	mostly frontal	close	108	Brazil
SSIG (test-set)	mostly frontal	medium,far	804	Brazil
AOLP (Road Patrol)	frontal + oblique	close	611	Taiwan
Proposed (CD-HARD)	mostly oblique	close,medium,far	102	Various

The more challenging dataset currently used in terms of LP distortion is the AOLP Road Patrol (RP) subset, which tries to simulate the case where a camera is installed in a patrolling vehicle or hand-held by a person. In terms of

[4] Available at https://github.com/openalpr/benchmarks.

distance from the camera to the vehicles, the SSIG dataset appears to be the most challenging one. It is composed of high-resolution images, allowing that LPs from distant vehicles might still be readable. None of them present LPs from multiple (simultaneous) vehicles at once.

Although all these databases together cover numerous situations, to the best of our knowledge there is a lack of more general-purpose dataset with challenging images in the literature. Thus, an additional contribution of this work is the manual annotation of a new set of 102 images (named as CD-HARD) selected from the Cars Dataset, covering a variety of challenging situations. We selected mostly images with strong LP distortion but still readable for humans. Some of these images (crops around the LP region) are shown in Fig. 1, which was used to motivate the problem tackled in this work.

4 Experimental Results

This section covers the experimental analysis of our full ALPR system, as well as comparisons with other state-of-the-art methods and commercial systems. Unfortunately, most academic ALPR papers focus on specific scenarios (e.g. single country or region, environment conditions, camera position, etc.). As a result, there are many scattered datasets available in the literature, each one evaluated by a subset of methods. Moreover, many papers are focused only on LP detection or character segmentation, which limits even more the comparison possibilities for the full ALPR pipeline. In this work, we used four independent datasets to evaluate the accuracy of the proposed method in different scenarios and region layouts. We also show comparisons with commercial products and papers that present full ALPR systems.

The proposed approach presents three networks in the pipeline, for which we empirically set the following acceptance thresholds: 0.5 for vehicle (YOLOv2) and LP (WPOD-NET) detection, and 0.4 for character detection and recognition (OCR-NET). Also, it is worth noticing that characters “I” and “1” are identical for Brazilian LPs. Hence, they were considered as a single class in the evaluation of the OpenALPR BR and SSIG datasets. No other heuristic or post-processing was applied to the results produced by the OCR module.

We evaluate the system in terms of the percentage of correctly recognized LPs, where an LP is considered correct if all characters were correctly recognized, and no additional characters were detected. It is important to note that the exact same networks were applied to all datasets: no specific training procedure was used to tune the networks for a given type of LP (e.g. European or Taiwanese). The only slight modification performed in the pipeline was for the AOLP Road Patrol dataset. In this dataset, the vehicles are very close to the camera (causing the vehicle detector to fail in several cases), so that we directly applied the LP detector (WPOD-NET) to the input images.

To show the benefits of including fully synthetic data in the OCR-NET training procedure, we evaluated our system using two sets training data: (i) real augmented data plus artificially generated ones; and (ii) only real augmented data.

Table 2: Full ALPR results for all 5 datasets.

	OpenALPR EU	OpenALPR BR	SSIG Test	AOLP RP	Proposed CD-HARD	Average
Ours	93.52%	91.23%	**88.56%**	**98.36%**	**75.00%**	**89.33%**
Ours (no artf.)	92.59%	88.60%	84.58%	93.29%	73.08%	86.43%
Ours (unrect.)	94.44%	90.35%	87.81%	84.61%	57.69%	82.98%
Commercial systems						
OpenALPR	**96.30%**	85.96%	87.44%	69.72%*	67.31%	81.35%
Sighthound	83.33%	**94.73%**	81.46%	83.47%	45.19%	77.64%
Amazon Rekog.	69.44%	83.33%	31.21%	68.25%	30.77%	56.60%
Literature						
Laroca et al. [17]	-	-	85.45%	-	-	-
Li et al. [18]	-	-	-	88.38%	-	-
Li et al. [19]	-	-	-	83.63%	-	-
Hsu et al. [10]	-	-	-	85.70%**	-	-

*OpenALPR struggled to understand the "Q" letter in Taiwanese LPs.

**In [10] the authors provided an estimative, and not the real evaluation.

These two versions are denoted by "Ours" and "Ours (no artf.)", respectively, in Table 2. As can be observed, the addition of fully synthetic data improved the accuracy in all tested datasets (with a gain $\approx 5\%$ for the AOLP RP dataset). Moreover, to highlight the improvements of rectifying the detection bounding box, we also present the results of using a regular non-rectified bounding box, identified as "Ours (unrect.)" in Table 2. As expected, the results do not vary much in the mostly frontal datasets (being even slightly better for ALPR-EU), but there was a considerable accuracy drop in datasets with challenging oblique LPs (AOLP-RP and the proposed CD-HARD).

Table 2 also shows the results of competitive (commercial and academic) systems, indicating that our system achieved recognition rates comparable to commercial ones in databases representing more controlled scenarios, where the LPs are mostly frontal (OpenALPR EU and BR, and SSIG). More precisely, it was the second best method in both OpenALPR datasets, and top one in SSIG. In the challenging scenarios (AOLP RP and the proposed CD-HARD dataset), however, our system outperformed all compared approaches by a significant margin (over 7% accuracy gain when compared to the second best result).

It is worth mentioning that the works of Li et al. [18,19], Hsu et al. [10] and Laroca et al. [17] are focused on a single region or dataset. By outperforming them, we demonstrate a strong generalization capacity. It is also important to note that the full LP recognition rate for the most challenging datasets (AOLP-RP and CD-HARD) was higher than directly applying the OCR module to the annotated rectangular LP bounding boxes (79.21% for AOLP-RP and 53.85% for CD-HARD). This gain is due to the unwarping allowed by WPOD-NET, which greatly helps the OCR task when the LP is strongly distorted. To illus-

trate this behavior, we show in Fig. 8 the detected and unwarped LPs for the images in Fig. 1, as well as the final recognition result produced by OCR-NET. The detection score of the top right LP was below the acceptance threshold, illustrating a false negative example.

Fig. 8: Detected/unwarped LPs from images in Fig. 1 and final ALPR results.

The proposed WPOD-NET was implemented using TensorFlow framework, while the initial YOLOv2 vehicle detection and OCR-NET were created and executed using the DarkNet framework. A Python wrapper was used to integrate the two frameworks. The hardware used for our experiments was an Intel Xeon processor, with 12Gb of RAM and an NVIDIA Titan X GPU. With that configuration, we were able to run the full ALPR system with an average of 5 FPS (considering all datasets). This time is highly dependent of the number of vehicles detected in the input image. Hence, incrementing the vehicle detection threshold will result in higher FPS, but lower recall rates.

5 Conclusions and Future Work

In this work, we presented a complete deep learning ALPR system for unconstrained scenarios. Our results indicate that the proposed approach outperforms existing methods by far in challenging datasets, containing LPs captured at strongly oblique views while keeping good results in more controlled datasets.

The main contribution of this work is the introduction of a novel network that allows the detection and unwarping of distorted LPs by generating an affine transformation matrix per detection cell. This step alleviates the burden of the OCR network, as it needed to handle less distortion.

As an additional contribution, we presented a new challenging dataset for evaluating ALPR systems in captures with mostly oblique LPs. The annotations for the dataset will be made publicly available so that the dataset might be used as a new challenging LP benchmark.

For future work, we want to extend our solution to detect motorcycle LPs. This poses new challenges due to differences in aspect ratio and layout. Moreover, we intend to explore the obtained affine transformations for automatic camera calibration problem in traffic surveillance scenarios.

Acknowledgements. The authors would like to thank the funding agencies CAPES and CNPq, as well as NVIDIA Corporation for donating a Titan X Pascal GPU.

References

1. Anagnostopoulos, C.N., Anagnostopoulos, I., Psoroulas, I., Loumos, V., Kayafas, E.: License Plate Recognition From Still Images and Video Sequences: A Survey. IEEE Transactions on Intelligent Transportation Systems **9**(3), 377–391 (sep 2008). https://doi.org/10.1109/TITS.2008.922938, http://ieeexplore.ieee.org/document/4518951/
2. Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M.: Segmentation- and Annotation-Free License Plate Recognition With Deep Localization and Failure Identification. IEEE Transactions on Intelligent Transportation Systems **18**(9), 2351–2363 (sep 2017). https://doi.org/10.1109/TITS.2016.2639020
3. Delmar Kurpiel, F., Minetto, R., Nassu, B.T.: Convolutional neural networks for license plate detection in images. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3395–3399. IEEE (sep 2017). https://doi.org/10.1109/ICIP.2017.8296912, http://ieeexplore.ieee.org/document/8296912/
4. Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. IEEE Transactions on Circuits and Systems for Video Technology **23**(2), 311–325 (feb 2013). https://doi.org/10.1109/TCSVT.2012.2203741, http://ieeexplore.ieee.org/document/6213519/
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (Jun 2010)
6. Gonalves, G.R., da Silva, S.P.G., Menotti, D., Schwartz, W.R.: Benchmark for license plate character segmentation. Journal of Electronic Imaging **25**(5), 1–5 (2016), http://www.ssig.dcc.ufmg.br/wp-content/uploads/2016/11/JEI-2016-Benchmark.pdf
7. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 4, pp. 770–778. IEEE (jun 2016). https://doi.org/10.1109/CVPR.2016.90
9. Hsu, G.S., Ambikapathi, A., Chung, S.L., Su, C.P.: Robust license plate detection in the wild. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. No. August, IEEE (aug 2017). https://doi.org/10.1109/AVSS.2017.8078493, http://ieeexplore.ieee.org/document/8078493/
10. Hsu, G.S., Chen, J.C., Chung, Y.Z.: Application-Oriented License Plate Recognition. IEEE Transactions on Vehicular Technology **62**(2), 552–561 (feb 2013). https://doi.org/10.1109/TVT.2012.2226218
11. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269. IEEE (jul 2017). https://doi.org/10.1109/CVPR.2017.243, http://arxiv.org/abs/1608.06993http://ieeexplore.ieee.org/document/8099726/
12. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/Accuracy

Trade-Offs for Modern Convolutional Object Detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3296–3297. IEEE (jul 2017). https://doi.org/10.1109/CVPR.2017.351, http://ieeexplore.ieee.org/document/8099834/

13. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. NIPS, Conference on Neural Information Processing Systems pp. 1–10 (2014)
14. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2017–2025. Curran Associates, Inc. (2015)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
16. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)
17. Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Gonçalves, G.R., Schwartz, W.R., Menotti, D.: A robust real-time automatic license plate recognition based on the YOLO detector. CoRR **abs/1802.09567** (2018), http://arxiv.org/abs/1802.09567
18. Li, H., Shen, C.: Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs. arXiv preprint arXiv:1601.05610 (jan 2016), http://arxiv.org/abs/1601.05610
19. Li, H., Wang, P., Shen, C.: Towards end-to-end car license plates detection and recognition with deep neural networks. CoRR **abs/1709.08828** (2017), http://arxiv.org/abs/1709.08828
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. pp. 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2
22. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 5 (2011)
23. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788. IEEE (jun 2016). https://doi.org/10.1109/CVPR.2016.91, http://ieeexplore.ieee.org/document/7780460/
24. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6517–6525. IEEE (jul 2017). https://doi.org/http://dx.doi.org/10.1109/CVPR.2017.690, http://arxiv.org/abs/1612.08242http://ieeexplore.ieee.org/document/8100173/
25. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (jun 2017). https://doi.org/10.1109/TPAMI.2016.2577031

26. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
27. Selmi, Z., Ben Halima, M., Alimi, A.M.: Deep Learning System for Automatic License Plate Detection and Recognition. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 1132–1138. IEEE (nov 2017). https://doi.org/10.1109/ICDAR.2017.187, http://ieeexplore.ieee.org/document/8270118/
28. Silva, S.M., Jung, C.R.: Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 55–62 (Oct 2017). https://doi.org/10.1109/SIBGRAPI.2017.14
29. Wang, F., Zhao, L., Li, X., Wang, X., Tao, D.: Geometry-Aware Scene Text Detection with Instance Transformation Network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1381–1389. Salt Lake City (2018)
30. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. IEEE Transactions on pattern analysis and machine intelligence **31**(10), 1733–1746 (2009)
31. Xie, L., Ahmad, T., Jin, L., Liu, Y., Zhang, S.: A New CNN-Based Method for Multi-Directional Car License Plate Detection. IEEE Transactions on Intelligent Transportation Systems **19**(2), 507–517 (feb 2018). https://doi.org/10.1109/TITS.2017.2784093
32. Zhou, X., Zhu, M., Leonardos, S., Daniilidis, K.: Sparse representation for 3d shape estimation: A convex relaxation approach. IEEE transactions on pattern analysis and machine intelligence **39**(8), 1648–1661 (2017)