IET Intelligent Transport Systems

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Advancing Multinational License Plate Recognition Through Synthetic and Real Data Fusion: A Comprehensive Evaluation

Rayson Laroca[1,2] | Valter Estevam[2,3] | Gladston J. P. Moreira[4] | Rodrigo Minetto[5] | David Menotti[2]

[1]Graduate Program in Informatics, Pontifical Catholic University of Paraná (PUCPR), Curitiba, Brazil | [2]Department of Informatics, Federal University of Paraná (UFPR), Curitiba, Brazil | [3]Federal Institute of Paraná (IFPR), Irati, Brazil | [4]Computing Department, Federal University of Ouro Preto (UFOP), Ouro Preto, Brazil | [5]Graduate Program in Electrical Engineering and Industrial Computing, Federal University of Technology-Paraná (UTFPR), Curitiba, Brazil

Correspondence: Rayson Laroca (rayson@ppgia.pucpr.br)

Received: 16 August 2024 | **Revised:** 30 June 2025 | **Accepted:** 13 August 2025

Funding: This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) and the National Council for Scientific and Technological Development (CNPq) (Grant Number: 315409/2023-1).

ABSTRACT

Automatic license plate recognition (ALPR) is a frequent research topic due to its wide-ranging practical applications. While recent studies use synthetic images to improve license plate recognition (LPR) results, there remain several limitations in these efforts. This work addresses these constraints by comprehensively exploring the integration of real and synthetic data to enhance LPR performance. We subject 16 optical character recognition (OCR) models to a benchmarking process involving 12 public datasets acquired from various regions. Several key findings emerge from our investigation. Primarily, the massive incorporation of synthetic data substantially boosts model performance in both intra- and cross-dataset scenarios. We examine three distinct methodologies for generating synthetic data: template-based generation, character permutation, and utilizing a generative adversarial network (GAN) model, each contributing significantly to performance enhancement. The combined use of these methodologies demonstrates a notable synergistic effect, leading to end-to-end results that surpass those reached by state-of-the-art methods and established commercial systems. Our experiments also underscore the efficacy of synthetic data in mitigating challenges posed by limited training data, enabling remarkable results to be achieved even with small fractions of the original training data. Finally, we investigate the trade-off between accuracy and speed among different models, identifying those that strike the optimal balance in each intra-dataset and cross-dataset settings.

1 | Introduction

Automatic license plate recognition (ALPR) systems employ image processing and pattern recognition techniques to locate and recognize license plates (LPs) in images or videos [1–3]. Traffic law enforcement, toll collection, and vehicle access control in restricted areas are some practical applications for an ALPR system [4, 5].

In the deep learning era, ALPR systems typically include two stages: license plate detection (LPD) and LPR [6–8]. The former refers to locating the LP regions in the input image, while the latter refers to identifying the characters on each LP. Recent research has focused on the LPR stage [9–11], as generic object detectors (e.g., YOLO) have achieved impressive results in the LPD stage for some time now [12–14].

Increased mobility and internationalization set new challenges for developing effective ALPR systems, as they must handle LPs from multiple regions with non-standardized formats [15, 16]. While ALPR systems have exhibited remarkable performance on LPs from diverse regions (e.g., Brazil, mainland China, Europe, Taiwan, among others) due to advances in deep learning and the increasing availability of annotated datasets [17–19], recent studies have indicated the existence of strong biases in ALPR research. An example worth mentioning is [20], where the authors showed that each dataset has a unique and identifiable "signature," as a lightweight classification model could predict the source dataset of an LP image at levels significantly better than chance.

One way to mitigate the problem of biases in automatic license plate recognition (ALPR) would be to embrace the "wildness" of the internet to collect a large-scale dataset from multiple sources [20, 21]. However, labeling such a dataset would be very expensive and time-consuming [22–24], not to mention the growing concerns surrounding privacy [25–27]. In this scenario, synthetic data emerges as a practical alternative, offering a cost-effective, privacy-preserving solution while providing the diversity and scale needed to train deep learning-based models effectively.

Although recent research has explored creating synthetic LP images to improve LPR performance, our analysis in Section 2 reveals certain limitations in these efforts. Existing studies have predominantly employed a single methodology to generate synthetic LPs, leaving unanswered questions regarding the potential for significantly enhanced outcomes by integrating data generated from various methodologies. Additionally, most works have focused on LPs from a single region, even though this limitation has been acknowledged for many years in the literature [28, 29]. To illustrate, researchers have trained separated instances of generative models—e.g., GANs—for different LP layouts. This approach becomes increasingly impractical and even unfeasible as the number of LP layouts the ALPR system must handle increases. Ultimately, the assessment of synthetic data generation methods has primarily relied on the performance of individual OCR models, overlooking the fact that images created using a particular method may disproportionately favor certain models over others.

This work aims to address the limitations described above by delving further into the integration of real and synthetic data to enhance LPR. Setting our research apart from previous studies, we subject 16 well-known OCR models to a benchmarking process across 12 public datasets acquired from multiple regions. Synthetic LP images are created by drawing inspiration from the three most widely adopted methodologies in the literature: a rendering-based pipeline (templates), character permutation, and a GAN model. We conduct ablation studies to demonstrate the impact of each methodology on the final results and the importance of synthetic data when training data is scarce.

In summary, this paper makes the following contributions:

- The most extensive experimental evaluation conducted in the field. While our focus lies on the LPR stage, as per recent research trends, we also compare various models for detecting the LPs and their corresponding corners within the input images. Our end-to-end experiments cover both intra- and cross-dataset evaluations, including an examination of the speed/accuracy trade-off of the OCR models;
- We deviate from prior methodologies by introducing a pipeline that employs a single GAN model to generate images of LPs from diverse regions and across styles. Notably, satisfactory outcomes are attained despite using a relatively small number of real images for training. This success stems from our approach of supplementing these real images with many synthetic ones created by a different method while also leveraging an OCR model to identify and filter out poorly generated images;
- Our results show that the massive use of synthetic data significantly improves the performance of the models, both in intra- and cross-dataset scenarios. Remarkably, employing the top-performing OCR model yields end-to-end results surpassing state-of-the-art methods and established commercial systems. These findings are particularly impressive because our models are not specifically trained for any particular LP layout, and we do not rely on post-processing with heuristic rules to improve the LPR performance on LPs from specific regions;
- Our ablation studies reveal that each synthesis method contributes considerably to enhancing the results, with a substantial synergistic effect observed when combining them. Incorporating synthetic data into the training set also proves to be effective in overcoming the challenges posed by limited training data, as commendable results are attained even when using only small fractions of the original data;
- We will publicly release all synthetic images created for training the OCR models, along with the accompanying code, thus enabling the generation of new images.

While recent studies have introduced new recognition architectures to boost LPR accuracy, our work adopts a complementary, data-centric approach. We show that significant performance improvements can be achieved across a wide range of architectures simply by enhancing the quality of the training data. This strategy supports fair, architecture-agnostic comparisons and emphasizes reproducibility and generalizability by providing plug-and-play training data that can be leveraged by any OCR model.

The remainder of this paper is structured as follows. Section 2 outlines the prevalent methods for synthesizing LP images in the literature. Section 3 elaborates on our methodology for generating synthetic data, which are integrated with real data to train the OCR models. Section 4 describes the experimental setup, including the datasets and models explored. The results are presented and analyzed in Section 5. Finally, Section 6 summarizes the key findings of this study.

2 | Related Work

In addition to the dataset bias mentioned in the previous section, LPR faces challenges related to unbalanced data. The inherent difficulty in collecting LP images from a variety of regions makes most ALPR datasets exhibit a significant bias towards specific regional identifiers [9, 30–32].

Considering the above discussion, many methods have been proposed to generate synthetic LP images. These methods aim to mitigate bias in the experiments and minimize the reliance on large volumes of real images for training OCR models. The subsequent paragraphs provide a concise overview of three widely adopted methods.

A highly intuitive approach for creating LP images involves a rendering-based process, particularly effective as LPs within a specific region typically conform to a strict standard. Put simply, such a method initiates with a blank template mirroring the actual aspect ratio and color scheme of LPs from the target region. Then, a random sequence of characters reflecting the actual LP sequence scheme is superimposed onto the template. Finally, transformations are applied to enhance the diversity of the generated images.

Several works have effectively explored the above methodology, including but not limited to [22, 24, 33]. Regarding the creation of LP images, these works primarily differed in the LP layout synthesized and the specific transformations applied. For instance, Björklund et al. [22] focused on creating Italian LPs, Maier et al. [33] generated German LPs, and Gao et al. [24] synthesized LPs from mainland China. In general, the transformations applied include modifications in font thickness, pixel shifts in character positions, LP rotation, adjustments in brightness and contrast, and the introduction of random shadows and noise.

Rendering-based methods face a significant limitation as they generate images with inconsistent distributions compared to real-world images, even when incorporating many transformations [24, 33, 34]. Consequently, LPR models trained solely on such images often produce unsatisfactory outcomes when applied to real-world images. Taking this into account, researchers have explored various approaches for creating realistic LP images, ranging from simpler methods such as character permutation to more complex strategies involving generative models (see below).

Generating synthetic data through character permutation is a simple yet effective method for achieving balance among character classes. Considering that each character's position on a given LP is labeled, one character can be replaced by another by superimposing the corresponding patch. Typically, this procedure focuses on replacing overrepresented characters in the training set with those underrepresented. To our knowledge, Gonçalves et al. [35] were the first to explore this permutation-based approach in the LPR context. Since then, several authors have successfully applied it to construct well-balanced training sets regarding character classes. The following paragraph presents three examples and briefly describes the subtle variations in how the respective authors implemented this method.

Shashirangana et al. [36] swapped character patches from distinct LP images, while most authors limited their permutations to character patches from the same LP to reduce illumination inconsistencies. Al-batat et al. [37] refrained from permuting patches of thin characters such as '1' and 'I' to prevent potential deformation caused by swapping them with wider characters. In contrast, other authors addressed this issue by first expanding the bounding boxes of smaller characters, incorporating portions of the LP background into them, to ensure uniform sizing of all characters before permutation. Lastly, although most authors swapped letters with digits and vice versa, Laroca et al. [38] only permuted letters with letters and digits with digits, enabling models to implicitly learn the fixed positions for letters and digits in certain LP layouts.

Concerning the use of generative models in LPR research, the prevailing choice has been GANs. The application of conditional GANs to image-to-image translation was first investigated by Isola et al. [39], with the proposal of the widely recognized pix2pix model. Pix2pix learns to map an image from the input to the output domain using an adversarial loss in conjunction with the L1 loss between the output and target images, thus requiring paired training data. While paired image-to-image translation models have shown remarkable results since this seminal work, acquiring such training data (i.e., matching image pairs with pixel-wise or patchwise labeling) can be time-consuming or even unrealistic. To tackle this challenge, CycleGAN, DualGAN and DiscoGAN provided a novel perspective (nearly simultaneously), in which the models discover relations between two visual domains without any explicitly paired data. As paired data is often unavailable, unpaired image-to-image translation has gained much attention in subsequent years. In the following paragraphs, we briefly describe recent publications that have employed GANs to generate synthetic data for improved LPR.

Wang et al. [31] employed CycleGAN [40] to transform a large number of script LP images, created using OpenCV, into realistic ones. Implementation details were not provided. Similarly, Zhang et al. [30] trained CycleGAN without the second cycle-consistency loss (i.e., they discarded the loss responsible for mapping real images into synthetic ones) to generate LP images with different characters and distinct characteristics. They trained multiple networks, each specialized in producing images with specific attributes. For instance, one model was trained to transform script images into bright LPs, while another was trained to convert script images into dark LPs, and so forth. In both works, LPs of only a few different styles (all from mainland China) were synthesized. Fan and Zhao [2] adopted essentially the same approach but trained CycleGAN with the Wasserstein distance loss. Their experiments focused solely on two distinct LP styles, one from mainland China and the other from the Taiwan region.

Han et al. [23] trained CycleGAN, StarGAN and pix2pix to generate images of the prevalent style of Korean LPs from script images. Their findings indicated that pix2pix produced more realistic and diverse LP images, supported by both qualitative comparisons and the superior performance of an OCR model trained with pix2pix-generated images compared to instances of the same model trained with images from CycleGAN and StarGAN. Shashirangana et al. [36] employed pix2pix to convert color images from the CCPD dataset into infrared images. They explored the KAIST multi-spectral dataset, which has 95k paired color and infrared images, for training the pix2pix model. The researchers suggested that the generated images could be employed to train an OCR model capable of identifying LPs extracted from real images captured during nighttime periods. Shvai et al. [32] built on several existing frameworks (e.g., AC-GAN and PG-GAN) to generate high-quality LP images with distinct sequences. In summary, their model achieves diversity by inputting the generator with different random latent vectors. It is worth noting that the authors focused on generating a single style of LPs, specifically the most common style found on vehicles in Texas, United States.

When examining the works described in this section, as well as others omitted for brevity, it becomes clear that the evaluation of methods for generating synthetic data has relied on the outcomes produced by individual OCR models. For example, Wang et al. [31] assessed the efficacy of their strategy solely based on the results achieved by their convolutional neural network (CNN)-based model. Similarly, Zhang et al. [30] considered only the results reached by an OCR model based on Xception, and Fan and Zhao [2] considered only the results yielded by CNNG, their multi-task model. We posit that such an evaluation is suboptimal because images created through a specific method may disproportionately benefit certain approaches over others. This phenomenon was evidenced in [41], where two segmentation-free models (Multi-Task and CRNN) had a much higher performance gain than the YOLO-based CR-NET model [42] when incorporating images generated via character permutation into the training set. Therefore, there is a lack of studies focused on evaluating these techniques' efficiency based on the results achieved by multiple OCR models with varying characteristics.

Another point that catches our attention is that most works still focus on LPs from a single region. In fact, it is not uncommon for only a very specific LP style (e.g., single-row blue LPs from mainland China) to be considered in the experiments [23, 32, 33]. Researchers have often opted to train separate instances of the proposed models for each LP layout. For example, one model generates/recognizes LPs from the Taiwan region, another model generates/recognizes LPs from mainland China, and so forth [22, 43, 44]. As mentioned earlier, this approach becomes increasingly impractical and even unfeasible as the number of LP layouts the ALPR system must handle increases. This impracticality arises from the necessity of adjusting parameters and retraining models whenever incorporating support for LPs from new regions or even different LP styles within the same region.

Ultimately, it is crucial to emphasize that within the examined literature, each work has exclusively generated synthetic LPs through a single methodology, such as relying solely on templates, employing only character permutation, or using GANs exclusively. It remains unclear whether relying on a single approach is sufficient for optimal results, or if considerably superior outcomes could be attained by integrating data generated through diverse methodologies. This study addresses this gap by benchmarking 16 OCR models across 12 publicly available datasets from different regions. We explore all three methodologies mentioned earlier for generating synthetic data, which are elaborated on in the following section. To our knowledge, this represents the most extensive experimental evaluation conducted in the ALPR field.

3 | Synthetic Data

This section details our approach to generate synthetic data, which is combined with real data to train the deep models for LPR. We start by outlining the methodology for creating LP images using blank templates and character patches sourced from the internet. Afterward, we delve into the process of producing new LP images by permuting the positions of the characters within each LP. Lastly, we elaborate on our utilization of a paired image-to-image translation model (pix2pix) to generate realistic LP images.

3.1 | Templates

While there are various approaches for creating LP images using templates, the method employed in this study is quite straightforward. First, blank templates that match the aspect ratio and color scheme of real LPs are sourced from the internet[1]. Subsequently, a sequence of characters, selected randomly yet crafted to mirror the patterns found on authentic LPs, is superimposed onto each template. Figure 1 shows examples of LP images generated through this process. Naturally, during the training of the OCR models, we subject these images to various transformations to introduce variability. These transformations encompass a range of techniques, including but not limited to random perspective transformation, introduction of random noise, incorporation of random shadows, and application of random changes to hue, saturation and brightness.

To better simulate real-world scenarios, the templates we generate using this method are derived from the LP styles observed within the training sets of the datasets explored in our experiments (see Section 4.1). In other words, we do not create templates for LP styles found exclusively in the test sets. To illustrate, one of the datasets we employ in our cross-dataset assessments contains images of electric vehicles registered in mainland China, which feature 8-character green LPs. Despite this, we refrain from creating templates for this LP style since it is not present in the training set.

An appealing aspect of this synthesis method lies in its ability to generate any sequence for each template while adhering to a predefined number of characters. Nevertheless, two drawbacks deserve attention. First, as highlighted in Section 2, images produced by such rendering-based approaches often exhibit inconsistent distributions compared to real-world images. Second, sourcing background and character images online for certain LP styles, particularly those less popular or recently introduced, can pose a challenge. This challenge plays a key role in our decision not to create templates for every LP style present in the training set, in addition to the inherent scope limitations of our study.

We generate 100k LP images employing this approach, a number determined through preliminary experiments that showed slightly improved outcomes compared to using 50k images and

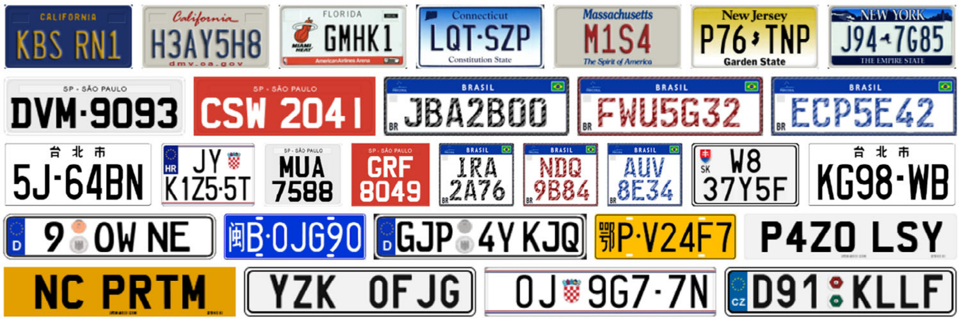


FIGURE 1 | Examples of the template-based LP images we created for this work. Notably, any sequence can be generated for each template. The background and character images were gathered from the internet [1]. During training, these LP images are subjected to various transformations to introduce variability.

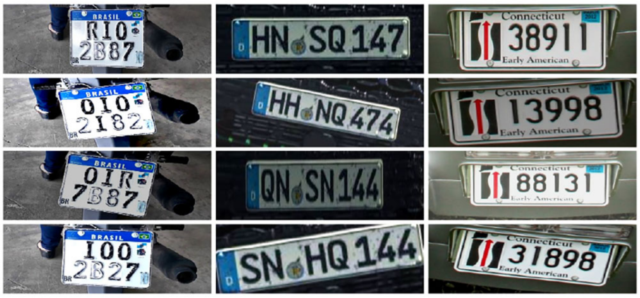

BRASIL R10 2B87
BRASIL 010 2182
BRASIL 01R 7B87
BRASIL 100 2B27
HN SQ147
HH NQ474
QN SN144
SN HQ144
Connecticut 38911 Early American
Connecticut 13998 Early American
Connecticut 88131 Early American
Connecticut 31898 Early American

FIGURE 2 | Some LP images created by permuting the positions of the characters within each LP and then applying transformations. The images in the top row are the originals, while the others were synthesized.

similar performance to using 200k images. The number of synthesized LPs is balanced across the six explored LP layouts (i.e., American, Brazilian, Chinese, European, Mercosur, and Taiwanese)[2], and the LP sequences are defined to maximize class balance for each character position.

3.2 | Character Permutation

Generating synthetic data through character permutation is also a straightforward process, outlined as follows. Initially, each character's bounding box (x, y, w, h) must be labeled. Then, if all the bounding boxes share the same width and height, the patch of each character can be replaced with another according to predefined rules. However, it is important to highlight that characters from distinct classes often differ in size, especially in terms of width. Adhering to established practices in the literature (refer to Section 2), we first expand the bounding boxes of smaller characters, incorporating small portions of the LP background into them, so that all characters have identical dimensions. Subsequently, we replace patches of characters that are overrepresented in the training set with patches from those that were underrepresented. To maintain consistency in illumination, we limit character permutation to patches within the same LP.

In Figure 2, we show examples of LP images generated by permuting the character positions on three LPs and applying random transformations of scale, rotation, brightness and cropping. Despite the impressive visual outcomes, it is essential to acknowledge certain limitations associated with this image synthesis method. First, manually labeling the bounding box for each character on every LP image is a laborious, time-consuming, and error-prone task [19, 22, 44]. Second, this method can only be applied to LP images where the character bounding boxes do not intersect (typically restricting its use on tilted LPs). Otherwise, parts of some characters may become obscured or replicated during the permutation process. Lastly, as the permutations involve repetitions and are limited to characters within the same LP, the OCR models may inadvertently learn undesirable correlations or biases. For instance, Gonçalves et al. [35] pointed out that characters from initially underrepresented classes exhibited a strong self-correlation, as they are more likely to appear in multiple positions on the permuted LPs (this is illustrated in Figure 2 as well).

We conducted a series of experiments in the validation set to determine the number of LP images to generate through this approach. We then generated 300k images, evenly distributed across the different LP layouts, as we found that generating a higher volume of images did not yield improved results.

3.3 | Image-To-Image Translation (pix2pix)

As outlined in Section 2, most previous works explored unpaired image-to-image translation methods (e.g., CycleGAN) to generate realistic LP images due to the lack of labeled paired data. In this work, we exploit the character permutation method described above to tackle this problem. More specifically, we generated over one million new LP images by shuffling the character positions on approximately 2k images from the training set of public datasets and the internet. While Laroca et al. [1] provided labels for most of these images, we further enriched the annotations by labeling the positions of the LP corners. The complete set of annotations will be publicly available.

Considering that these images are accompanied by precise annotations for the position of each LP corner and the bounding box of every character, they can be used to train paired image-to-image translation methods. In this study, we employ the renowned pix2pix model [39] for synthesizing many realistic images of LPs from multiple regions. We remark that although there are newer models available that would certainly yield better results than pix2pix, our decision to opt for pix2pix is primarily based on its widespread availability across various

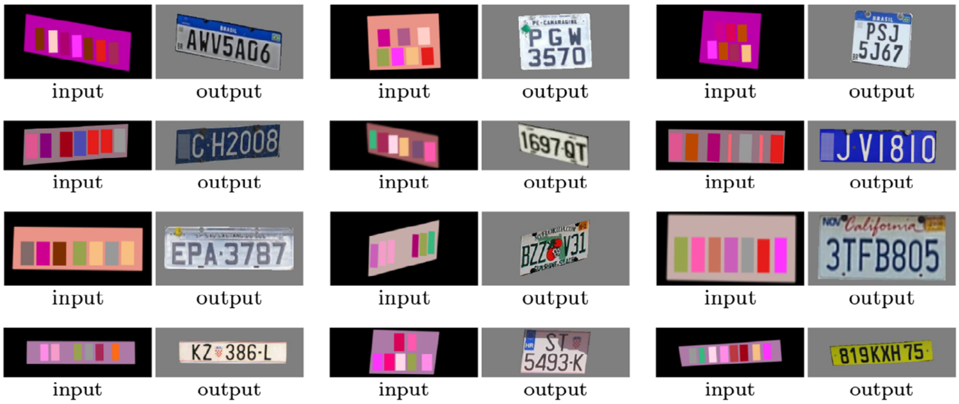


FIGURE 3 | Examples of image pairs used for training the pix2pix model. To create the input masks, labels are required for both the LP's layout and corners, as well as for the bounding box of each character.

frameworks such as Chainer, Keras, PyTorch, TensorFlow, Torch, and others[3]. This choice is particularly significant for our research, given that part of our experiments were conducted on an old CPU lacking AVX instructions, significantly limiting the available framework options.

The paired data required for training the pix2pix model is prepared as follows. For each LP image generated through character permutation, which serves as the intended output, a corresponding segmentation mask is created to serve as the input. These masks are designed such that each color represents a distinct LP layout class or character class. For example, as shown in Figure 3, the digit '0' is indicated by a vivid red color (228, 28, 26), the letter 'A' is denoted by a dark brown shade (126, 47, 0), the Mercosur layout is represented by a purplish-magenta tone (187, 0, 170), and the Chinese layout is denoted by a gray color (127, 127, 127). The Glasbey library[4] was employed to generate a set of colors that are maximally distinguishable from each other. Black (0, 0, 0) and similar shades are avoided in this process since black in the input mask represents the background. Notably, the background in the output LP image consists of gray pixels. This choice was made because using the original background led to inferior results.

After completing the model's training, the next step involves using it to generate hundreds of thousands of new LP images. Intuitively, this task is accomplished by feeding the model with segmentation masks derived from randomly selected LP layouts and character sequences. While the characters are sampled from the valid alphabet per position, we ensure a balanced distribution of character classes at every position.

Upon examining the generated LP images, we discovered that although many high-quality LPs were produced, a notable portion of them also displayed certain issues. The primary issue identified was the distortion of characters or their blending into two distinct classes. For instance, a generated character might exhibit a fusion of traits from '0' and '8', with the defining strokes that typically differentiate the two appearing faint and indistinct. To address this matter, we ran the Fast-OCR model, which demonstrated superior cross-dataset results among a dozen recognition models in [38], on the millions of generated images and selected the top N predictions according to their associated confidence values. Specifically, we selected the top 50k images for each of the six explored LP layouts, totaling 300k images. This strategy proved effective in filtering out most images with defects, although it may have excluded some instances with a higher degree of variability. Examples of the selected images are shown in Figure 4.

It should be noted that we train the pix2pix model to produce a blurred representation instead of Chinese characters (this can be seen in Figures 3, 4). This adjustment is made due to the absence of class labels for these characters in the training set. Accurately labeling these characters poses a challenging task for individuals not proficient in Chinese, which is the case for our team. Further details on how we handle Chinese characters in our experiments can be found in Section 4.3.

One might question the rationale behind employing segmentation maps as input for the pix2pix model, rather than using LP templates. While we acknowledge that using templates as input would likely yield similar or even better results, the lack of LP style-related annotations in public datasets poses a challenge. The provided information is limited to the geographical region where the images were collected (e.g., Europe, mainland China, and the United States). Fundamentally, adopting LP templates as input would entail labeling the specific style of each LP and searching online platforms for the corresponding templates and character patches (or creating them using OpenCV or similar tools). This is most likely why previous works explored very few LP styles in their experiments [2, 30, 31].

The major limitation of this method stems from its reliance on the training data, as it cannot synthesize LP layouts that are not included in the training set [24].

4 | Experimental Setup

This section describes the experimental setup adopted in this work. We first present the datasets explored, elucidating their division into training, validation and test subsets. Subsequently, we list the OCR models we implemented for our assessments, providing the rationale for their choice over alternative options. Lastly, we detail how the performance evaluation is conducted.

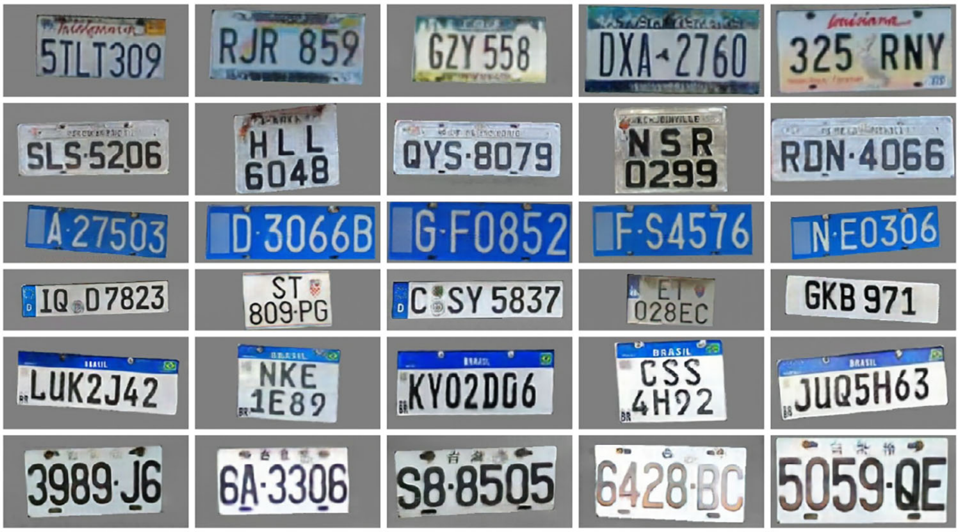


FIGURE 4 | Examples of selected images from those generated using pix2pix. From top to bottom, we show American, Brazilian, Chinese, European, Mercosur, and Taiwanese LPs.

TABLE 1 | The 12 datasets used in our experiments.

Dataset	Images	Resolution	LP layout
Caltech Cars [45]	126	896 × 592	American
EnglishLP [46]	509	640 × 480	European
UCSD-Stills [47]	291	640 × 480	American
ChineseLP [48]	411	Various	Chinese
AOLP [49]	2049	Various	Taiwanese
OpenALPR-EU[a] [50]	108	Various	European
SSIG-SegPlate [51]	2000	1920 × 1080	Brazilian
PKU[a] [52]	2253	1082 × 727	Chinese
UFPR-ALPR [53]	4500	1920 × 1080	Brazilian
CD-HARD[a] [54]	102	Various	Various
CLPD[a] [30]	1200	Various	Chinese
RodoSol-ALPR [55]	20000	1280 × 720	Brazilian & Mercosur

[a]Datasets used only for testing the deep models (i.e., cross-dataset experiments).

While different machines were used for model training, all testing experiments were conducted on a PC equipped with an AMD Ryzen Threadripper 1920X 3.5GHz CPU, 96 GB of RAM running at 2133 MHz, an SSD with read and write speeds of 3500 MB/s and 3000 MB/s, respectively, and an NVIDIA Quadro RTX 8000 GPU (48 GB).

4.1 | Datasets

As shown in Table 1, our experiments were conducted on images from 12 well-known public datasets gathered over the past two decades across distinct regions. Figure 5 presents representative LP images from these datasets, highlighting the variability in LP layouts and visual characteristics. As detailed in the following paragraph, eight of these datasets were used for training, validation, and testing of the selected models, while the remaining four datasets were reserved exclusively for testing (cross-dataset experiments).

For reproducibility, here we detail how we split the images from each dataset into training, validation and test sets[5]. The UCSD-Stills, SSIG-SegPlate, UFPR-ALPR and RodoSol-ALPR datasets were divided according to the protocols defined by the respective authors. The other datasets, which do not have standard protocols, were split following prior studies. More specifically, the images from the OpenALPR-EU, PKU, CD-HARD and CLPD datasets were used exclusively for testing, as in [18, 44, 56]. The Caltech Cars dataset was randomly divided into 63.5% of the images for training/validation and 36.5% for testing, as in [57–59]. Following Panahi & Gholampour [60], Henry et al. [17], Beratoğlu & Töreyin [61], we randomly split the EnglishLP dataset as follows: 80% of the images for training/validation and 20% for testing. Regarding the ChineseLP dataset, we followed the protocol adopted by Laroca et al. [1, 38]: 40% of the images for training, 20% for validation, and 40% for testing[6]. Lastly, we divided each of the three subsets of the AOLP dataset (i.e., AC, LE, and RP) into training and test sets with a 2:1 ratio, as Xie et al. [13], Zhuang et al. [62], Liang et al. [63], and used 20% of the training images for validation.

To ensure a minimum of 500 training images for each LP layout, we expanded our training set with 772 images from the internet. These images were labeled and made available by Laroca et al. [1] and encompass 257 American LPs, 347 Chinese LPs, and 178 European LPs. Furthermore, to address potential overfitting issues, we employed Albumentations [64]—a popular image augmentation library—to balance the number of training images from different datasets.

We opted not to explore the CCPD dataset [7] in our experiments, despite its widespread use in the literature. There are two primary reasons for this decision. First, the dataset comprises highly compressed images, significantly reducing the legibility of the

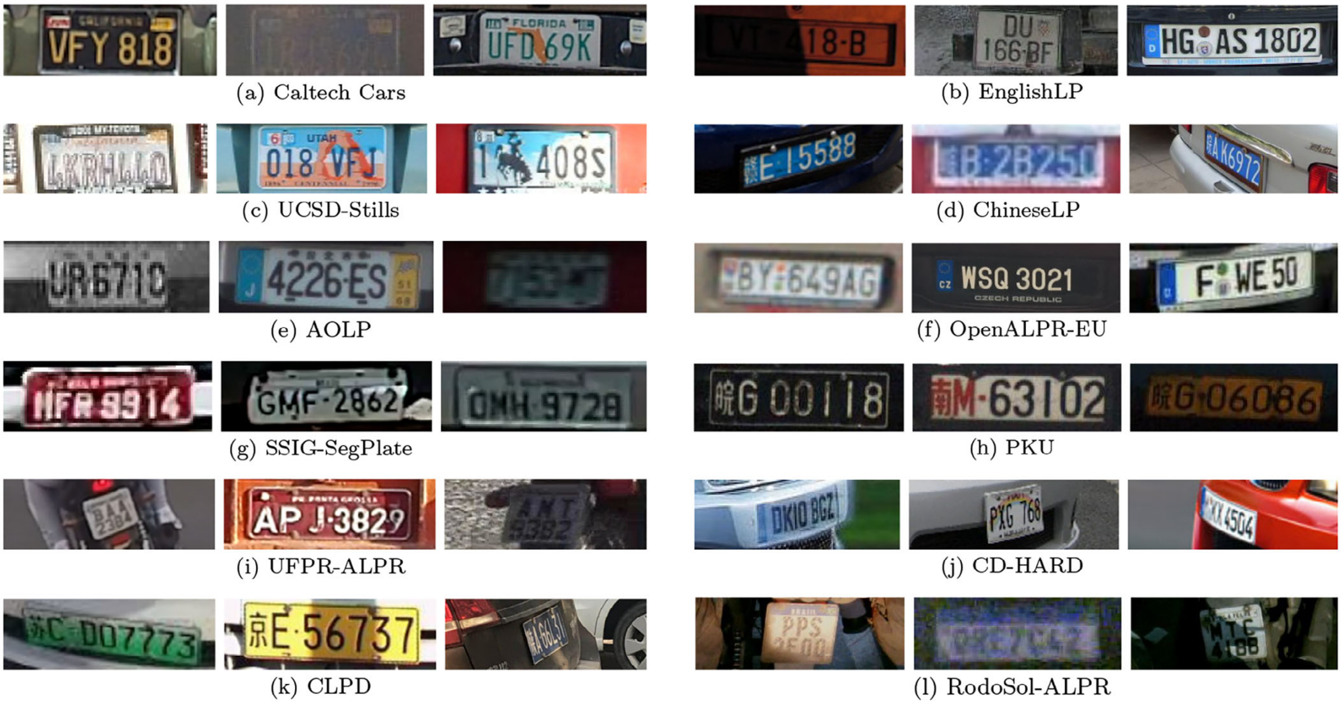

(a) Caltech Cars
(b) EnglishLP
(c) UCSD-Stills
(d) ChineseLP
(e) AOLP
(f) OpenALPR-EU
(g) SSIG-SegPlate
(h) PKU
(i) UFPR-ALPR
(j) CD-HARD
(k) CLPD
(l) RodoSol-ALPR

FIGURE 5 | Representative LPs from the datasets used in our experiments, highlighting the diversity of LP layouts and visual characteristics. Adapted from [38].

LPs [18], and this does not align with our intended application. Qiao et al. [65] even observed that some images within CCPD are too blurry for the LPs to be recognized. Second, the CCPD dataset has experienced multiple updates and expansions since its introduction. Consequently, there is an inconsistency regarding the dataset's size across different studies. While some sources claim it contains 250k images [2, 5, 63], others suggest a range of 280-290k images [24, 44, 66], whereas the current version has 366,789 images. The divergence in test sets across different versions renders the results reported in various studies not directly comparable.

TABLE 2 | The 16 OCR models explored in our experiments.

Model	Original application
Framework: PyTorch[9]	
R^2AM [67]	Scene text recognition
RARE [68]	Scene text recognition
STAR-Net [69]	Scene text recognition
CRNN [70]	Scene text recognition
GRCNN [71]	Scene text recognition
Rosetta [72]	Scene text recognition
TRBA [73]	Scene text recognition
ViTSTR-Base [74]	Scene text recognition
ViTSTR-Small [74]	Scene text recognition
ViTSTR-Tiny [74]	Scene text recognition
Framework: Keras[8]	
Holistic-CNN [75]	License plate recognition
Multi-Task [35]	License plate recognition
Multi-Task-LR [76]	License plate recognition
CNNG [2]	License plate recognition
Framework: Darknet[7]	
CR-NET [42]	License plate recognition
Fast-OCR [77]	Image-based meter reading

4.2 | OCR Models

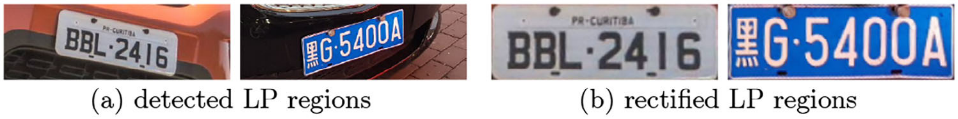
This work compares 16 OCR models applied to the LPR task. Table 2 presents an overview of these models, listing the original application for which they were designed and specifying the framework we used to implement them.

We selected these models for two primary reasons. First, they have a proven track record of success in OCR tasks (including but not limited to LPR) [2, 10, 18, 73, 74, 78]. Second, we are confident in our ability to train and adjust them effectively to ensure fairness in our experiments, as the authors provided enough details about the model architectures, and also because we designed/employed similar networks in previous works [35, 38, 76, 77]. We are unaware of any work in the ALPR literature where so many OCR models were explored in the experiments.

Each model was trained and tested using either the framework in which it was originally implemented or well-known public repositories associated with it. In summary, the YOLO-based models (i.e., CR-NET and Fast-OCR) were implemented using the Darknet framework[7]; the multi-task models (those listed in the middle section of Table 2) were implemented using Keras[8];

TABLE 3 | Results obtained by YOLOv4-CSP and IWPOD-NET in the LP detection stage (@ IoU > 0.7). For this evaluation, the corners predicted by IWPOD-NET were converted into bounding boxes.

Model	Metric	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
YOLOv4-CSP	Recall	100.0%	99.0%	100.0%	98.1%	99.9%	100.0%	99.2%	100.0%	**99.5%**
IWPOD-NET		95.7%	100.0%	100.0%	97.5%	99.7%	98.8%	82.4%	99.6%	96.7%
YOLOv4-CSP	Precision	100.0%	97.1%	96.8%	98.1%	94.8%	94.9%	97.8%	99.6%	**97.4%**
IWPOD-NET		66.7%	77.9%	73.2%	83.1%	88.3%	61.6%	62.2%	78.4%	73.9%
YOLOv4-CSP	F-score	100.0%	98.1%	98.4%	98.1%	97.3%	97.5%	98.5%	99.8%	**98.5%**
IWPOD-NET		81.2%	88.9%	86.6%	90.3%	94.0%	80.2%	72.3%	89.0%	85.3%


(a) detected LP regions

(b) rectified LP regions

FIGURE 6 | Two LPs before and after the rectification process. Observe that the rectified LPs resemble frontal views, becoming more horizontal, tightly bounded, and easier to read.

and the other models were implemented using a popular fork of the open source repository of Clova AI Deep Text Recognition Benchmark[9].

The hyperparameters used for training the models were defined based on preliminary experiments carried out in the validation set and are as follows. In Darknet, we employed the stochastic gradient descent (SGD) optimizer, 65k iterations, batch size = 64, and learning rate = [10^{-3}, 10^{-4}, 10^{-5}] with decay steps at 40% and 70% of the total iterations. In Keras, we used the Adam optimizer, learning rate = 10^{-5}, batch size = 64, max epochs = 100, and patience = 7 (the number of epochs with no improvement after which training is stopped). In PyTorch, we adopted the following parameters: Adadelta optimizer (decay rate $\rho = 0.95$), 300k iterations, and batch size = 128.

4.3 | Performance Evaluation

We present the models' performance for each dataset by calculating the ratio of correctly recognized LPs to the number of LPs in the test set. We remark that an LP is deemed correctly recognized only if all its characters are precisely identified, given that a single misidentified character can lead to the misidentification of the vehicle.

As mentioned in Section 1, recent research has placed emphasis on the LPR stage [10, 11, 19, 24]. However, in our experiments, the LP patches fed into the recognition models were not cropped and rectified directly from the ground truth. Instead, we detected the LPs in the original images using YOLOv4-CSP [79] and rectified them through a combination of CDCC-NET [77]—for locating the LP corners—and perspective transformation. We adopted this procedure to fairly compare our results with end-to-end ALPR systems and to more accurately simulate real-world scenarios, where the LPs are not always optimally detected.

We chose YOLOv4-CSP over more recent models such as YOLOv10 and YOLOv11 for three practical and interconnected reasons. First, it is implemented in a fast and mature C/C++ version of the Darknet framework, supporting real-time processing in resource-constrained environments. Second, it already delivers strong performance in our experiments, with an F-score of 98.5% at IoU > 0.7 across all datasets (see Table 3 in Section 5.1), which we consider sufficient for the purposes of this study. Third, its outputs are directly utilized by CDCC-NET to locate the LP corners used in the rectification step—a crucial process that helps align LPs and reduce the impact of minor detection errors (the rectification process is detailed in the next paragraph). Although we acknowledge that more recent YOLO versions generally offer higher detection accuracy, YOLOv4-CSP meets both the accuracy and efficiency requirements of our pipeline.

We rectify each LP by calculating and applying a perspective transform from the coordinates of the four corners in the detected LP region to the corresponding vertices in the "unwarped" image. These corresponding vertices were defined as follows: (0, 0) corresponds to the top-left corner; ($max_w - 1$, 0) is the top-right corner; ($max_w - 1$, $max_h - 1$) refers to the bottom-right corner; and (0, $max_h - 1$) indicates the bottom-left corner, where max_w denotes the maximum distance between the top-right and top-left x coordinates or the bottom-right and bottom-left x coordinates, and max_h is the maximum distance between the top-left and bottom-left y coordinates or the top-right and bottom-right y coordinates. The rectification process is illustrated in Figure 6. Recent works that exploited LP rectification to improve the recognition results include [2, 18, 44, 80, 81].

It is essential to highlight that, in our experiments, we refrained from using prior knowledge about individual LP layouts to enhance the results through post-processing. As an illustration, despite being aware that all LPs in a given dataset or particular region adhere to a fixed pattern (e.g., Brazilian LPs are composed

TABLE 4 | Corner detection results achieved by four models within the regions found by YOLOv4-CSP. The results are presented in terms of LP-NME, where lower values indicate higher accuracy.

Test set # LPs Model	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
LocateNet [84]	0.0739	0.0359	0.0782	0.1092	0.0730	0.0329	0.0556	0.0592	0.0647
Hybrid-MobileNetV2 [85]	0.0323	0.0226	0.0352	0.0391	0.0332	0.0214	0.0313	0.0383	0.0317
IWPOD-NET [18]	0.0244	0.0143	0.0205	0.0138	0.0205	0.0098	0.0194	0.0141	0.0171
CDCC-NET [77]	0.0160	0.0117	0.0164	0.0176	0.0142	0.0098	0.0168	0.0150	**0.0147**

of three letters followed by four digits), we treat the predictions made by the models as final. We argue that by exposing the models to sufficient variability in the training stage, they can, to varying extents, implicitly learn and leverage such information to yield better predictions.

We follow the methodology adopted by Li et al. [82] and Laroca et al. [55], where all Chinese characters are collectively represented as a unified class denoted by '*'. Accordingly, all results from other studies presented in our comparison with the state of the art (Section 5.2.2) were obtained in the same way, disregarding Chinese characters.

5 | Results and Discussion

This section presents and analyzes the outcomes of our experiments. Section 5.1 offers a concise overview of the results obtained in detecting the LPs and locating the corresponding corners. The precise detection of the LP corners is pivotal for accurately rectifying the LPs before recognition. Section 5.2 then delves into a detailed examination of the end-to-end results obtained by employing different OCR models.

5.1 | LP Detection and Corner Detection

Various quantitative criteria can be employed to evaluate detection tasks. Our assessment includes the widely adopted Precision, Recall and F-score metrics. In line with prior studies [7, 8, 81], we consider the detections correct when the intersection over union (IoU) with the ground truth exceeds 0.7. Detections that meet this threshold typically encompass all LP characters.

Table 3 presents the results obtained by YOLOv4-CSP [79] and IWPOD-NET [18] (a well-known model specifically designed for LP detection). Three key observations can be drawn from the results: (i) YOLOv4-CSP demonstrated satisfactory results, both in terms of Precision and Recall, with instances of slightly lower Precision attributed to unlabeled LPs in the background of frames; (ii) while IWPOD-NET directly predicts LP corners rather than bounding boxes, its performance is suboptimal in scenarios where the vehicles are far from the camera, as evidenced by the Recall rates reached in the UFPR-ALPR dataset; and (iii) IWPOD-NET tends to predict a significant number of false positives, leading to notably low Precision rates. Despite our exploration of higher detection thresholds, doing so led to the exclusion of many true LPs (lower Recall rates). These observations likely influenced the decision of [18] to feed regions identified by a vehicle detector (YOLOv3) into IWPOD-NET instead of applying it directly to the original image (vehicle detection was not investigated in this study because most datasets lack labels for vehicle bounding boxes). Balancing Precision and Recall is crucial for an efficient system operation, as it relies on accurately detecting all LPs while minimizing false positives.

To rectify the LPs found by YOLOv4-CSP, it is necessary to locate the four corners associated with each of them. Table 4 presents a comparison of the results obtained in this process by four models specifically designed for corner detection, including IWPOD-NET. The evaluation is carried out in terms of LP-NME [83], a metric inspired by normalization mean error (NME), which in turn is commonly employed to evaluate the quality of face alignment algorithms. LP-NME is defined as follows:

$$\text{LP-NME}(C, \hat{C}) = \frac{1}{4} \sum_{i=1}^{4} \frac{||C_i - \hat{C}_i||}{d}, \tag{1}$$

where C and $\hat{C}$ are the ground truth and predicted corners, respectively, and d is the normalization factor. Following [83], we adopt the diagonal length of the smallest bounding box that completely encloses the LP as the normalization factor.

CDCC-NET stands out as the top-performing model, achieving the lowest average LP-NME value of 0.0147. It is noteworthy, however, that the IWPOD-NET model outperformed CDCC-NET in two datasets and achieved near-identical results in another. Figure 7 showcases the predictions made by all models for five distinct LP images. Although there is an evident similarity in the predictions for some LPs, the CDCC-NET model exhibits superior overall accuracy.

The findings outlined in this section substantiate our choice to employ YOLOv4-CSP for LP detection and CDCC-NET for corner detection. As elaborated in Section 4.3, the corners predicted by CDCC-NET are used to rectify the LPs before recognition.

5.2 | Overall Evaluation (End-To-End)

This section conducts a thorough comparative analysis of the OCR models, assessing their performance and contrasting the end-to-end results attained when employing the top-performing

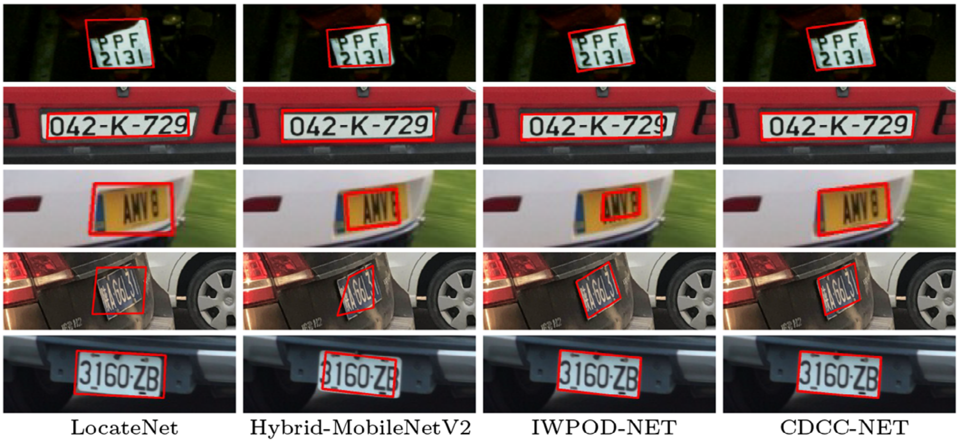


FIGURE 7 | Representative qualitative results achieved by four different models in corner detection. For better viewing, we draw a polygon from the predicted corner positions.

model with those reached by state-of-the-art approaches and established commercial systems (Sections 5.2.1, 5.2.2, 5.2.3). Notably, the evaluation covers both intra- and cross-dataset scenarios. Additionally, ablation studies are incorporated to demonstrate the impact of each explored method for generating synthetic images on the final results, as well as the importance of synthetic data when training data is scarce. Finally, Section 5.2.4 examines the trade-off between speed and accuracy exhibited by the recognition models, highlighting those that strike a favorable balance.

5.2.1 | Intra-Dataset Experiments

Table 5 presents the end-to-end results obtained across the disjoint test sets of the eight datasets used for training and validating the models. In these experiments, all OCR models were trained using real images combined with synthetic ones generated by the three methods described in Section 3. Later in this section, we present an ablation study that details the contribution of each image synthesis method to the results achieved.

The first observation is that all models performed surprisingly well, reaching average recognition rates between 94.6% and 97.9%. It is noteworthy that the mean results were well above 90% across all datasets, including UFPR-ALPR, which is known to be quite challenging [5, 86, 87]. According to our analysis of the results (presented throughout this section), such impressive results are mainly due to the massive use of synthetic data combined with the LP rectification stage.

Another point that immediately draws attention is that multiple models achieved the best result in at least one dataset. For instance, the CNNG excelled in the UFPR-ALPR dataset, while the Multi-Task-LR and Holistic-CNN models reported the highest recognition rates on ChineseLP and RodoSol-ALPR, respectively. Interestingly, the models that performed better on average (i.e., STAR-Net and TRBA) did not achieve the best results in six of the eight datasets; some models actually reached the best result in one dataset and the worst in another (e.g., see the results achieved by the CNNG and Holistic-CNN models on the EnglishLP dataset). These results emphasize the importance of evaluating and comparing OCR models on various datasets.

Figure 8 showcases the predictions yielded by the STAR-Net and TRBA models for LPs with distinct characteristics. The outcomes underscore the models' robustness in handling diverse LP layouts, images with varying resolutions, LPs with different numbers of characters arranged in one or two rows, and scenarios where the characters are partially occluded. Impressively, some of these LP styles were not even included in the training set. Overall, errors are limited to instances where one character closely resembles another, often due to factors such as low resolution and artifacts on the LP. Although this qualitative analysis focuses on the two models that achieved the best average results across the datasets, the other models generally produced similar predictions.

An important aspect to highlight is the effectiveness of synthetic data in scenarios with limited training samples—a common issue in public datasets collected from specific regions. Table 6 reports the average recognition rates of STAR-Net and TRBA when trained on progressively smaller subsets (50%, 25%, 10%, 5% and 1%) of the original training set, comprising images from the eight datasets listed in Section 4.1, both with and without the addition of synthetic data generated as described in Section 3. Remarkably, incorporating synthetic data in the training phase enabled commendable results to be reached even when using small fractions of the original training set. For example, both STAR-Net and TRBA achieved an average recognition rate exceeding 94.5% across all datasets when trained with only 10% of the original training set but supplemented with synthetic data. In contrast, relying solely on real images with common transformations as data augmentation led to a substantial decline in the results. Specifically, the recognition rates dropped below 75% when halving the original training set and plummeted to approximately 1% when using only 10% of it. This underscores the effectiveness of synthetic data in mitigating the challenges posed by limited training data.

Table 7 elucidates the effectiveness of each image synthesis method described in Section 3, as well as their combination, to the results obtained. It reveals that each method contributes consid-

TABLE 5 | Recognition rates obtained by all models under the intra-dataset protocol, where each model was trained once on the union of the training set images from these datasets (plus synthetic data) and evaluated on the respective test sets. The best results achieved in each dataset are shown in bold.

Model \ Test set # LPs	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
CNNG [2]	**97.8%**	91.2%	96.7%	98.8%	99.1%	98.8%	**96.1%**	97.1%	96.9%
CR-NET [42]	93.5%	96.1%	98.3%	96.9%	98.7%	98.0%	89.3%	88.3%[a]	94.9%
CRNN [70]	93.5%	96.1%	96.7%	95.7%	98.8%	97.5%	87.0%	92.2%	94.7%
Fast-OCR [77]	95.7%	97.1%	95.0%	96.9%	98.7%	96.0%	89.6%	88.1%[a]	94.6%
GRCNN [71]	**97.8%**	**99.0%**	96.7%	98.8%	99.0%	97.9%	87.4%	93.0%	96.2%
Holistic-CNN [75]	95.7%	91.2%	93.3%	99.4%	99.3%	98.4%	94.9%	**97.9%**	96.3%
Multi-Task [35]	**97.8%**	94.1%	**100.0%**	98.8%	99.1%	98.6%	93.3%	95.1%	97.1%
Multi-Task-LR [76]	95.7%	93.1%	93.3%	**100.0%**	99.6%	97.5%	94.6%	96.6%	96.3%
R^{2}AM [67]	**97.8%**	94.1%	95.0%	98.8%	99.3%	99.3%	90.6%	94.4%	96.1%
RARE [68]	**97.8%**	97.1%	98.3%	98.1%	99.4%	99.1%	91.9%	96.5%	97.3%
Rosetta [72]	95.7%	98.0%	98.3%	98.1%	98.7%	98.3%	92.6%	96.0%	97.0%
STAR-Net [69]	**97.8%**	**99.0%**	98.3%	98.1%	99.1%	99.3%	94.7%	97.0%	**97.9%**
TRBA [73]	**97.8%**	**99.0%**	98.3%	98.8%	98.8%	99.3%	94.0%	97.3%	**97.9%**
ViTSTR-Base [74]	95.7%	96.1%	93.3%	99.4%	**99.9%**	**99.4%**	94.6%	97.7%	97.0%
ViTSTR-Small [74]	95.7%	96.1%	98.3%	98.1%	99.1%	98.5%	94.9%	96.8%	97.2%
ViTSTR-Tiny [74]	93.5%	95.1%	91.7%	98.8%	99.0%	98.9%	92.3%	95.3%	95.5%
Average	96.2%	95.8%	96.4%	98.3%	99.1%	98.4%	92.4%	94.9%	96.4%

[a] Images from the RodoSol-ALPR dataset were not used for training the CR-NET and Fast-OCR models, as each character's bounding box needs to be labeled for training them.

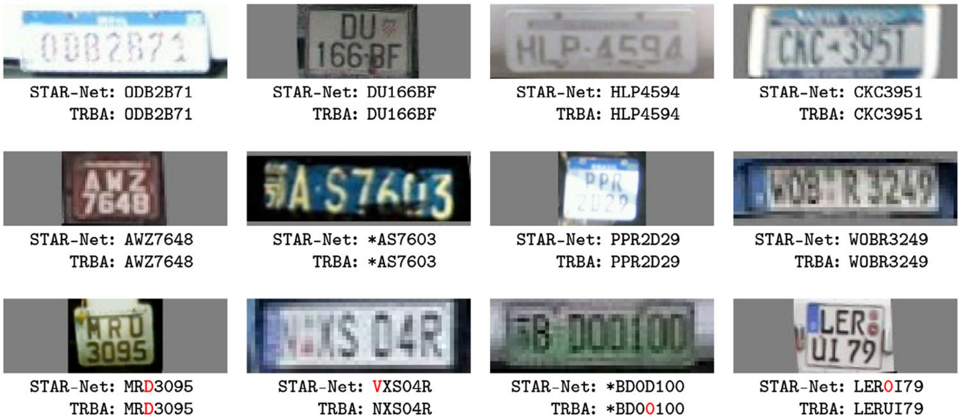


FIGURE 8 | Predictions made for 12 LP images by STAR-Net and TRBA, the two models that exhibited the highest average performance in the intra-dataset experiments. Errors, if any, are highlighted in red. All LPs are well aligned because they were rectified before recognition, as detailed in Section 4.3.

TABLE 6 | Average recognition rates obtained by STAR-Net and TRBA when trained with reduced portions of the original training data. Naturally, images not included in the reduced training set were not used to generate synthetic images in the respective experiments.

Model \ Real images	100%	50%	25%	10%	5%	1%
STAR-Net (no synthetic)	95.3%	62.0%	18.3%	1.3%	0.2%	0.0%
STAR-Net (w/ synthetic)	97.9%	95.8%	94.7%	94.6%	93.6%	86.4%
TRBA (no synthetic)	93.7%	74.0%	23.9%	0.9%	0.2%	0.0%
TRBA (w/ synthetic)	97.9%	97.0%	96.0%	94.5%	94.3%	87.9%

TABLE 7 | Average recognition rates and corresponding standard deviations obtained across all models and datasets with different types of images included in the training set. The values shown below each image synthesis method represent the number of images used for that method. "Data aug." refers to online data augmentation (using standard transformations) applied to real images. The synergistic impact of the three image synthesis methods in enhancing the overall results is evident. As creating synthetic images through character permutation and GAN relies on the existence of real images, their integration is evaluated only in scenarios where real images are included in the training set.

Real Images 15k (+ data aug)	Templates 100k	Permutation 300k	GAN 300k	Average	Average (rect.)
	✓			42.5% ± 13.9%	46.5% ± 11.5%
✓				84.5% ± 10.5%	88.1% ± 8.1%
✓		✓		91.4% ± 3.6%	93.6% ± 2.3%
✓	✓			92.5% ± 2.7%	94.7% ± 1.3%
✓			✓	93.2% ± 2.1%	95.2% ± 1.5%
✓	✓	✓		93.8% ± 1.6%	95.5% ± 0.8%
✓		✓	✓	94.0% ± 1.6%	95.6% ± 1.2%
✓	✓		✓	94.1% ± 1.7%	95.8% ± 1.0%
✓	✓	✓	✓	**94.9% ± 1.6%**	**96.4% ± 1.1%**

erably to enhancing the results. Notably, a substantial synergistic effect is observed when combining these methods, pushing the performance boundaries of recognition models applied to LPR. To elaborate, the best recognition rates (i.e., 94.9% and 96.4% for unrectified and rectified LPs, respectively), on average for all models, were achieved by combining original data with images synthesized in all three ways. To further support these findings, paired t-tests confirmed statistically significant differences ($p < 0.01$) between the top-performing configurations in both setups.

When real images were combined solely with images generated through character permutation, as in [36, 38], the average recognition rates obtained were 91.4% and 93.6% for unrectified and rectified LPs, respectively. Combining real images with LP templates alone, as in [24, 33], resulted in average recognition rates of 92.5% and 94.7% for unrectified and rectified LPs, respectively. Finally, the combination of real images with those generated through a GAN model (in our case, pix2pix), as in [30, 32], yielded average recognition rates of 93.2% and 95.2% for unrectified and rectified LPs, respectively. Note that training with LP templates alone, without any real images, resulted in much lower performance, with average recognition rates below 50%, likely due to the significant domain difference between these synthetic images and real LPs.

It is important to highlight how much better the results were when training the models with both real and synthetic images (i.e., 94.9% and 96.4%) compared to those obtained when simply training the models with original images augmented by common transformations such as random rotation, random noise, random cropping, random compression, and random changes in brightness, saturation, and contrast (i.e., 84.5% and 88.1%).

It is also noteworthy that both the templates and the images produced by the GAN model contributed significantly more to improving the OCR models' performance than the images generated through character permutation. This observation aligns with the fact that the images created via character permutation share many characteristics with their original counterparts (e.g., character position, compression artifacts, camera noise, among others) despite having different character sequences.

5.2.2 | Cross-Dataset Experiments

As the performance of LP recognition under the traditionally adopted intra-dataset protocol—where models are trained and tested on disjoint subsets of the same dataset—is rapidly improving, many researchers argue that a more realistic assessment comes from cross-dataset experiments. Such experiments better reflect real deployments where new cameras are deployed more frequently than models are retrained [11, 20, 44, 55]. Accordingly, Table 8 presents the results obtained by the same models used for the intra-dataset evaluation (Table 5), now tested on four entirely unseen datasets. No adjustments were made to the models, ensuring a faithful representation of their cross-dataset performance.

These results demonstrate that the explored OCR models, trained on a combination of real and synthetic images, maintain high performance even in unseen scenarios. What most caught our attention was the consistency of the TRBA model [73], as it also reached the best results in this evaluation. On the other hand, here the STAR-Net model (which tied with the best results in the intra-dataset experiments) was outperformed by RARE in all datasets. Based on these findings, we consider the configuration combining *YOLOv4-CSP* for detection, *CDCC-NET* for rectification, and *TRBA* for recognition as the most effective setup in our benchmark. Accordingly, we adopt this configuration for comparisons with state-of-the-art approaches in the next section.

While subpar results were achieved on the CD-HARD dataset, it is essential to recognize the inherent complexity of this dataset, as implied by its name. Our analysis has revealed that the primary challenge posed by this dataset lies in the diverse range of LP layouts it encompasses. Images within the dataset feature vehicles from various regions not represented in the datasets used for

TABLE 8 | Recognition rates obtained by all models on four public datasets that were not seen during the training stage (cross-dataset experiments). The best results for each dataset are shown in bold.

Dataset # LPs Model	OpenALPR-EU # 108	PKU # 2,253	CD-HARD # 104	CLPD # 1,200	Average
CNNG [2]	95.4%	98.6%	58.7%	92.9%	86.4%
CR-NET [42]	93.5%	**99.5%**	67.3%	92.9%	88.3%
CRNN [70]	97.2%	99.1%	56.7%	94.2%	86.8%
Fast-OCR [77]	98.1%	99.1%	69.2%	94.4%	90.2%
GRCNN [71]	97.2%	99.0%	57.7%	94.5%	87.1%
Holistic-CNN [75]	95.4%	99.0%	54.8%	94.0%	85.8%
Multi-Task [35]	96.3%	98.8%	54.8%	93.7%	85.9%
Multi-Task-LR [76]	94.4%	98.8%	53.8%	92.6%	84.9%
R^{2}AM [67]	98.1%	99.4%	57.7%	93.8%	87.3%
RARE [68]	**99.1%**	99.1%	72.1%	95.2%	91.4%
Rosetta [72]	97.2%	99.2%	64.4%	93.8%	88.7%
STAR-Net [69]	98.1%	98.5%	71.2%	95.0%	90.7%
TRBA [73]	**99.1%**	99.4%	**76.9%**	**96.2%**	**92.9%**
ViTSTR-Base [74]	94.4%	99.0%	54.8%	93.4%	85.4%
ViTSTR-Small [74]	96.3%	97.4%	59.6%	94.3%	86.9%
ViTSTR-Tiny [74]	94.4%	97.6%	53.8%	92.3%	84.5%
Average	96.5%	98.8%	61.5%	93.9%	87.7%

model training, such as Dubai and New South Wales. The high degree of tilt of many LPs would further hinder recognition if not rectified before the recognition stage [18, 54].

5.2.3 | Comparison With Previous Works and Commercial Systems

In Table 9, we compare the end-to-end results obtained by the best-performing model combination in our benchmark with those reported by state-of-the-art ALPR systems. Following common practice, to make the comparison fair, we only consider systems evaluated in the same way as in our benchmark (see details in Section 4.1). We also compare our results with those obtained by [88] and [89], which are two commercial systems widely adopted as baselines in the literature [4, 15].

It is impressive that, without using any heuristic rules or post-processing, the benchmark's top-performing setup (using TRBA for recognition) achieves state-of-the-art performance on all datasets except AOLP. Note that higher recognition rates (e.g., 99.9%) were actually attained on the AOLP dataset using other OCR models (see Table 5); however, we do not consider those results here because the respective models did not outperform TRBA on average across the full benchmark.

Two other aspects should be highlighted from the above results. First, the positive influence of exploiting synthetic data is reaffirmed, as our system did not achieve the best results on most datasets when solely using real data (plus simple data augmentation) for training. Second, both the [88] and [89] commercial systems performed poorly on the RodoSol-ALPR dataset (with 57.0% and 69.3% recognition rates, respectively). As shown in Table 10, the primary reason for such underwhelming results is that these systems do not work well for motorcycle LPs (which are challenging in nature, having two rows of characters and being smaller in size) or Mercosur LPs (which were adopted just a few years ago). These observations underscore the importance of comparing ALPR systems across diverse datasets that encompass various collection methodologies, feature images of different types of vehicles (including motorcycles), and exhibit different LP layouts (including two-row configurations).

There are many recent works where the authors evaluated the generalizability of the proposed methods in the PKU [52] and CLPD [30] datasets, both collected in mainland China. Hence, in Table 11, we compare the results obtained by these methods (plus Sighthound and OpenALPR) with those reached by the best-performing configuration in our benchmark. For each approach, we also report the number of real Chinese LPs used during training and indicate whether the method qualifies as multinational, defined here as not being trained or fine-tuned exclusively on Chinese LPs.

When exploring synthetic data for training the OCR model, the end-to-end approach (YOLOv4-CSP + CDCC-NET + TRBA) exhibited significantly superior performance compared to state-of-the-art methods and commercial systems on both datasets. These results are particularly noteworthy given that our training dataset comprised only 506 real images of vehicles with Chinese LPs, while most baseline models were trained on over 100,000 images from the CCPD dataset [7]. Indeed, this is one of the

TABLE 9 | Recognition rates obtained by our best approach (which uses TRBA as the recognition model), state-of-the-art methods, and two commercial systems in the eight datasets where part of the images was used for training the networks. The best results achieved in each dataset are shown in bold.

Test set / Approach	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
[88]	87.0%	93.1%	96.7%	95.0%	95.5%	82.8%	62.9%	57.0%	83.7%
[90][b]	91.3%	–	**98.3%**	–	–	–	–	–	–
[18]	–	–	–	–	97.4%	–	86.3%	–	–
[17]	**97.8%**	97.1%	–	–	98.9%	–	–	–	–
[1] (run 1)[a]	**97.8%**	96.1%	96.7%	98.1%	**99.4%**	98.8%	89.7%	–	–
[87]	–	–	–	–	–	–	90.3%	–	–
[18][a]	–	–	–	–	99.0%	–	91.8%	–	–
[38]	87.0%	88.2%	86.7%	96.9%	**99.4%**	95.8%	89.7%	95.6%	92.4%
[89][a]	95.7%	98.0%	**98.3%**	96.9%	97.1%	93.0%	92.2%	69.3%	92.6%
[56]	–	–	–	–	–	–	–	96.6%	–
[10][b]	–	–	–	–	–	–	–	96.6%	–
Ours	87.0%	91.2%	88.3%	98.1%	98.4%	98.1%	92.1%	96.8%	93.7%
[86]	–	–	–	–	–	98.6%	92.3%	–	–
[59][b]	–	–	–	–	99.0%	–	–	97.0%	–
Ours + synthetic	**97.8%**	**99.0%**	**98.3%**	**98.8%**	98.8%	**99.3%**	**94.0%**	**97.3%**	**97.9%**

[a] ALPR systems that rely on pre-defined heuristic rules (prior knowledge) to refine the predictions returned by the OCR model.
[b] The LP patches fed into the OCR model were cropped directly from the ground truth in [10, 59, 90].

TABLE 10 | Results achieved by two well-known commercial systems in the RodoSol-ALPR dataset. It can be seen that their capabilities vary considerably according to the vehicle type and the LP layout.

	Vehicle type		LP layout	
System	Cars	Motorcycles	Brazilian	Mercosur
[88]	81.3%	32.7%	63.9%	50.1%
[89]	95.6%	43.0%	90.7%	47.8%

reasons why this configuration did not outperform the baselines even further, especially on the CLPD dataset, as several of the recognition errors occurred on LP styles missing in our training set but present in CCPD (e.g., 8-character green LPs from electric vehicles). To assess the upper-bound performance in a comparable setup, we conducted an additional experiment by incorporating LP images from CCPD's training set into the training data, consistent with previous studies. In this setting, the benchmark configuration achieved recognition rates of 97.3% on CLPD and 99.5% on PKU, demonstrating its strong generalization ability when trained with a more comprehensive dataset.

5.2.4 | Speed/Accuracy Trade-Off

The importance of devising methods that strike an optimal balance between speed and accuracy has been highlighted in recent ALPR research [5, 8, 81]. Thus, this section examines the speed/accuracy trade-off of the OCR models explored in this study. Figure 9 compares the average recognition rates reached across datasets and the corresponding frames per second (FPS) processing capabilities of all models, both in intra- and cross-dataset setups.

In *intra-dataset* scenarios, the multi-task models, particularly Multi-Task and CNNG, demonstrated an exceptional balance between speed and accuracy. This can be attributed to their ability to learn potential classes for each character position independently, thereby avoiding confusion between similar letters and digits in layouts where they appear in distinct positions. When the primary goal is to achieve the utmost recognition rate across various scenarios, STAR-Net is a more compelling option than TRBA. This is due to STAR-Net achieving the same average recognition rate as TRBA (97.9%) while processing more than twice the FPS (141 vs. 59).

In *cross-dataset* scenarios, as outlined in Section 5.2.2, TRBA once again emerged as the top performer in terms of average recognition rate, standing alone this time, while STAR-Net was outperformed by RARE. Concerning the trade-off between speed and accuracy, the Fast-OCR model clearly excels, striking a commendable balance between the two. Its relatively high accuracy on unseen LPs can be attributed to its foundation on the YOLO object detector. Consequently, it detects and recognizes each character individually, as opposed to predicting specific LP sequences that mimic patterns from the training set. Conversely, the multi-task models experienced a substantial decline in recognition rate precisely because they learned to predict sequences based on

TABLE 11 | Comparison of the recognition rates (%) obtained by our best approach (TRBA), state-of-the-art methods, and commercial systems on the CLPD and PKU datasets. These experiments assess the generalizability of these ALPR approaches, as no images from those datasets were used for training. The methods categorized as "Multinational" were not trained or fine-tuned exclusively on Chinese LPs. Additional experiments show that our pipeline further outperforms others even when trained using the CCPD dataset, highlighting its effectiveness in both low-resource and large-scale scenarios.

Approach	Real images of Chinese LPs used for training	Multinational	Recognition CLPD	Rate PKU
[88]	?	✓	85.2%	89.3%
[30]	100,000+		87.6%	90.5%
[2]	100,000+	✓	88.5%	92.5%
Ours	506	✓	90.1%	96.8%
[3][a]	4,444		91.4%	96.1%
[9]	10,000		91.7%	–
[89]	?		91.8%	96.0%
[56]	100,000+		92.4%	92.8%
[8]	100,000+		93.2%	–
[66]	100,000+		94.0%	96.6%
[91]	100,000+		94.5%	–
[31]	100,000+		94.8%	–
[44]	100,000+		95.3%	96.9%
Ours + synthetic	**506**	✓	**96.2%**	**99.4%**
[Additional experiments]				
Ours + CCPD's training set	100,000+	✓	94.5%	96.8%
Ours + CCPD's training set + synthetic	**100,000+**	✓	**97.3%**	**99.5%**

[a] Approaches in which we applied the authors' code and pre-trained models to obtain the reported results.

patterns observed in the training set, which often differ from those observed in other datasets/scenarios.

Regarding the ViTSTR variants, it is worth noting that they handle essentially the same number of FPS. This is because the key differentiation among the ViTSTR-Base, -Small and -Tiny models lies in their respective number of parameters and computations required (FLOPS), rather than in the number of FPS they can process [74].

6 | Conclusions

This paper delves into the integration of real and synthetic data for improved LPR. Synthetic LP images were generated using three widely adopted methodologies in the literature: a rendering-based pipeline (templates), character permutation, and a GAN model. We subjected 16 OCR models to a thorough benchmarking process involving 12 public datasets acquired from various regions. The experiments encompassed both intra- and cross-dataset evaluations, including an examination of the speed/accuracy trade-off of the models. To the best of our knowledge, this constitutes the most extensive experimental evaluation conducted in the field.

Several key findings emerge from our study. Primarily, the massive use of synthetic data significantly improves the performance of all models, both in intra- and cross-dataset scenarios. The quantitative and qualitative results demonstrated the models' robustness in effectively handling diverse LP layouts, images with varying resolutions, and LPs with varying numbers of characters arranged in one or two rows. Notably, employing the top-performing OCR model (TRBA) yielded end-to-end results that surpassed those reached by state-of-the-art methods and established commercial systems. These results are particularly noteworthy as our models were not specifically trained for each LP layout, and we refrained from incorporating heuristic rules to enhance the predictions for LPs from specific regions through post-processing—a departure from many existing methods. This streamlined approach significantly simplifies the process of incorporating support for LPs from new regions or even markedly different LP styles within the same region.

The conducted ablation studies provide two important insights. First, each synthesis method contributes considerably to enhancing the results, and a substantial synergistic effect is observed when they were combined. This finding contrasts with the common practice of generating synthetic LPs exclusively through a single methodology. Second, incorporating synthetic data into the training set enables commendable results to be attained even when using small fractions of the original data. This highlights the effectiveness of synthetic data in overcoming the challenges posed by scarce training data.

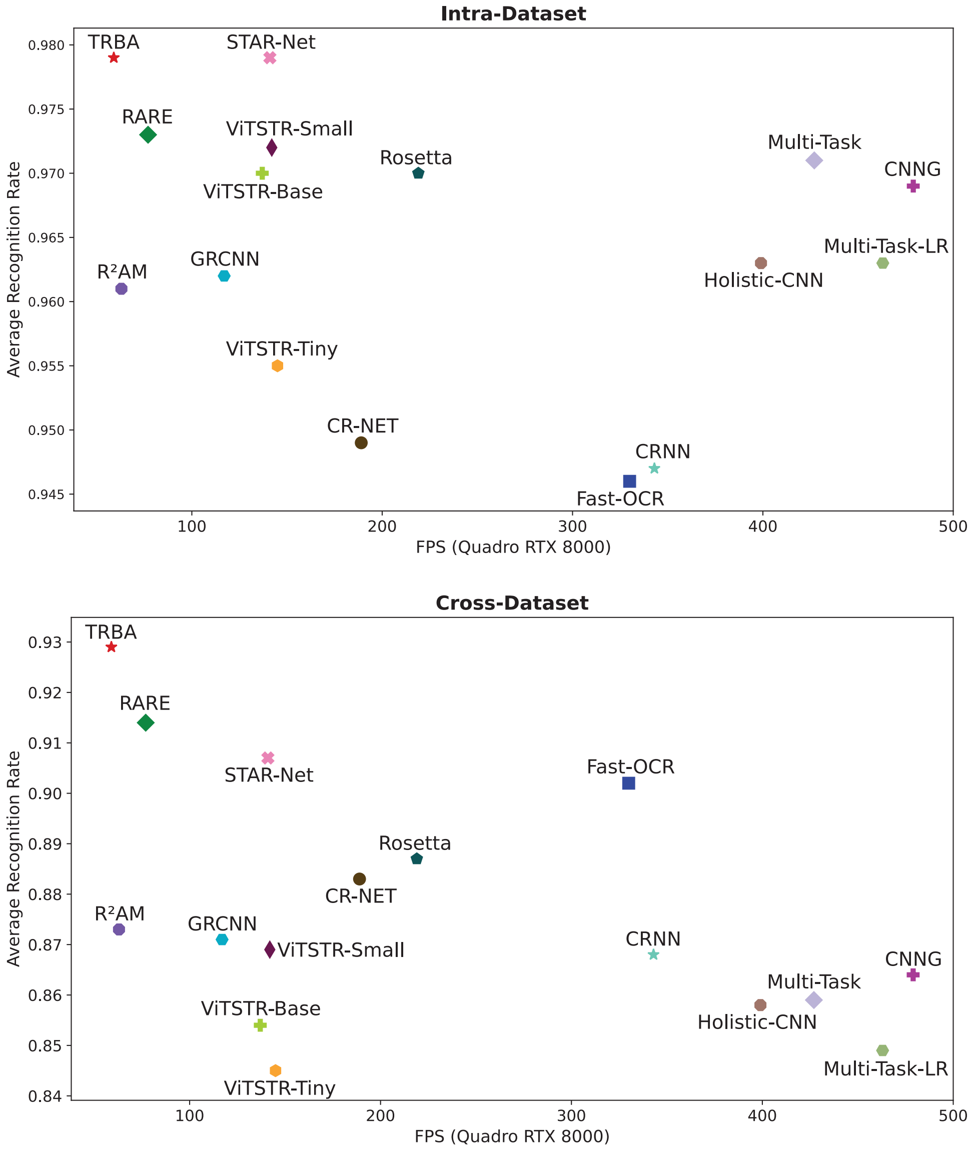


FIGURE 9 | Average recognition rate across datasets and the corresponding FPS processing capabilities for all OCR models on intra-dataset (top) and cross-dataset (bottom) experiments. The specific FPS value for each model is as follows: CNNG: 479; CR-NET: 189; CRNN: 343; Fast-OCR: 330; GRCNN: 117; Holistic-CNN: 399; Multi-Task: 427; Multi-Task-LR: 463; R^{2}AM: 63; RARE: 77; Rosetta: 219; STAR-Net: 141; TRBA: 59; ViTSTR-Base: 137; ViTSTR-Small: 142; and ViTSTR-Tiny: 145.

Acknowledging the significance of both model speed and accuracy in real-world applications, we investigated how well the models strike a balance between these two factors. Although the multi-task models demonstrated an impressive speed/accuracy trade-off in intra-dataset scenarios, this optimal balance did not extend to cross-scenario scenarios. In such instances, these models exhibited a more substantial decline in recognition rates than most other models. Remarkably, in cross-dataset scenarios, Fast-OCR stood out due to its great balance between speed and accuracy. The effectiveness of Fast-OCR in cross-dataset scenarios can be attributed to its character-level detection and recognition approach, setting it apart from other models that predict LP sequences by replicating patterns from the training set. While this replication approach proves effective in similar contexts, its efficacy tends to diminish when applied to different regions or scenarios.

An additional noteworthy contribution of this work lies in our commitment to granting access to all synthetic images generated for training the models. We will also provide the corresponding code for generating new LP images using each synthesis method.

It is essential to acknowledge the extensive number of experiments conducted for this study. We carried out nine training sessions for each of the 16 recognition models under investigation (refer to Table 7), subjecting them to testing across various seen and unseen datasets. We also explored the pix2pix model's capabilities for generating LP images and performed multiple experiments related to the LP detection and corner detection tasks, as reported in Tables 3, 4. Note that a single training process for some models (e.g., TRBA and ViTSTR-Base) took several days to complete on an NVIDIA Quadro RTX 8000 GPU, which is currently one of the top-performing GPUs in the market.

6.1 | Future Directions

Despite the comprehensive scope of our study and the significantly greater dataset diversity compared to prior work, we recognize an important area for improvement regarding the geographic, script, and layout variability of LPs. In particular, our evaluation did not include datasets with non-Latin scripts, such as Arabic or Cyrillic, nor LPs with radically different formats, such as vertical or non-standard layouts. These omissions may affect the global applicability of our findings and highlight the need for future research to assess synthetic data generation methods and OCR models in regions with alternative scripts and diverse LP layouts.

Building on the demonstrated effectiveness of synthetic data for accurately recognizing LPs in high-quality images captured under diverse conditions and across multiple regions, we advocate for a gradual shift in the focus of ALPR research. Specifically, there is a pressing need to address the challenges posed by low-resolution or low-quality LPs. These difficult scenarios—often encountered in criminal investigations where the LP remains unclear across all video frames—remain significantly underexplored in the current literature.

Future work should also consider pairing our synthetic data pipeline with architecture-level innovations, such as neural architecture search or lightweight adapters, to further improve recognition performance, especially in unconstrained environments.

Although synthetic data is widely regarded as a privacy-preserving solution, we acknowledge that its use in ALPR systems does not eliminate broader ethical concerns, such as the potential for misuse in large-scale surveillance or data manipulation. In addition to providing an ethical use guideline in the public code repository accompanying this work, we encourage future research to investigate methods that improve traceability and accountability in synthetic data pipelines. In particular, exploring mechanisms such as data watermarking and auditing tools is essential to support responsible use and reduce the risk of misuse in sensitive surveillance contexts.

Author Contributions

Rayson Laroca: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft. **Valter Estevam:** conceptualization, investigation, writing – original draft, writing – review & editing. **Gladston J. P. Moreira:** conceptualization, investigation, writing – review & editing. **Rodrigo Minetto:** conceptualization, investigation, writing – review & editing, supervision. **David Menotti:** conceptualization, investigation, writing – review & editing, supervision, resources, project administration, funding acquisition.

Acknowledgements

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) and the National Council for Scientific and Technological Development (CNPq) (Grant Number: 315409/2023-1). The Quadro RTX 8000 GPU used for this research was donated by the NVIDIA Corporation.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available at https://raysonlaroca.github.io/supp/lpr-synthetic-data/.

Endnotes

[1] Most of the blank templates and character patches were taken from https://platesmania.com/.

[2] As in previous works, the "Chinese" layout refers to LPs assigned to vehicles registered in mainland China, while the "Taiwanese" layout denotes LPs issued for vehicles registered in the Taiwan region.

[3] See a list of pix2pix implementations at https://phillipi.github.io/pix2pix/. Our chosen implementation can be found at https://github.com/affinelayer/pix2pix-tensorflow.

[4] https://github.com/taketwo/glasbey.

[5] The complete list of which images from each dataset were used for training, validation and testing can be downloaded at https://raysonlaroca.github.io/supp/lpr-synthetic-data/splits.zip

[6] To train the models, we excluded the few images from the ChineseLP dataset that are also found in CLPD (both datasets include internet-sourced images, as discussed by Laroca et al. [92]).

[7] https://github.com/AlexeyAB/darknet/.

[8] https://keras.io/.

[9] https://github.com/roatienza/deep-text-recognition-benchmark/.

References

1. 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," *IET Intelligent Transport Systems* 15, no. 4 (2021): 483–503, https://doi.org/10.1049/itr2.12030.

2. X. Fan and W. Zhao, "Improving Robustness of License Plates Automatic Recognition in Natural Scenes," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 10 (2022): 18845–18854, https://doi.org/10.1109/TITS.2022.3151475.

3. Z. Rao, D. Yang, N. Chen, and J. Liu, "License Plate Recognition System in Unconstrained Scenes via a New Image Correction Scheme and Improved CRNN," *Expert Systems with Applications* 243 (2024): 122878, https://doi.org/10.1016/j.eswa.2023.122878.

4. W. Weihong and T. Jiaoyang, "Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment," *IEEE Access* 8 (2020): 91661–91675, https://doi.org/10.1109/ACCESS.2020.2994287.

5. H. Ding, J. Gao, Y. Yuan, and Q. Wang, "An End-to-End Contrastive License Plate Detector," *IEEE Transactions on Intelligent Transportation Systems* 25, no. 1 (2024): 503–516, https://doi.org/10.1109/TITS.2023.3304816.

6. H. Li, P. Wang, M. You, and C. Shen, "Reading Car License Plates Using Deep Neural Networks," *Image and Vision Computing* 72 (2018): 14–23, https://doi.org/10.1016/j.imavis.2018.02.002.

7. Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," in *European Conference on Computer Vision (ECCV)* (Springer, 2018), 261–277, https://doi.org/10.1007/978-3-030-01261-8_16.

8. X. Ke, G. Zeng, and W. Guo, "An Ultra-Fast Automatic License Plate Recognition Approach for Unconstrained Scenarios," *IEEE Transactions*

on Intelligent Transportation Systems 24, no. 5 (2023): 5172–5185, https://doi.org/10.1109/TITS.2023.3237581.

9. Q. Liu, S.-L. Chen, Z.-J. Li, C. Yang, F. Chen, and X.-C. Yin, "Fast Recognition for Multidirectional and Multi-Type License Plates with 2D Spatial Attention," in *International Conference on Document Analysis and Recognition (ICDAR)* (Springer, 2021), 125–139, https://doi.org/10.1007/978-3-030-86337-1_9.

10. V. Nascimento, R. Laroca, J. A. Lambert, W. R. Schwartz, and D. Menotti, "Super-Resolution of License Plate Images Using Attention Modules and Sub-Pixel Convolution Layers," *Computers & Graphics* 113 (2023): 69–76, https://doi.org/10.1016/j.cag.2023.05.005.

11. F. Schirrmacher, B. Lorch, A. Maier, and C. Riess, "Benchmarking Probabilistic Deep Learning Methods for License Plate Recognition," *IEEE Transactions on Intelligent Transportation Systems* 24, no. 9 (2023): 9203–9216, https://doi.org/10.1109/TITS.2023.3278533.

12. G. S. Hsu, A. Ambikapathi, S. L. Chung, and C. P. Su, "Robust License Plate Detection in the Wild," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (Aug 2017), 1–6, https://doi.org/10.1109/AVSS.2017.8078493.

13. L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A New CNN-Based Method for Multi-Directional Car License Plate Detection," *IEEE Transactions on Intelligent Transportation Systems* 19, no. 2 (Feb 2018): 507–517, https://doi.org/10.1109/TITS.2017.2784093.

14. X. Hu, H. Li, X. Li, and C. Wang, "MobileNet-SSD MicroScope Using Adaptive Error Correction Algorithm: Real-Time Detection of License Plates on Mobile Devices," *IET Intelligent Transport Systems* 14, no. 2 (2020): 110–118, https://doi.org/10.1049/iet-its.2019.0380.

15. Lubna, N. Mufti, and S. A. A. Shah, "Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms," *Sensors* 21, no. 9 (2021): 3028, https://doi.org/10.3390/s21093028.

16. T.-A.-L. Trinh, T.-A. Pham, and V.-D. Hoang, "Layout-Invariant License Plate Detection and Recognition," in *International Conference on Multimedia Analysis and Pattern Recognition* (2022), 1–6, https://doi.org/10.1109/MAPR56351.2022.9924802.

17. C. Henry, S. Y. Ahn, and S. Lee, "Multinational License Plate Recognition Using Generalized Character Sequence Detection," *IEEE Access* 8 (2020): 35 185–35 199, https://doi.org/10.1109/ACCESS.2020.2974973.

18. S. M. Silva and C. R. Jung, "A Flexible Approach for Automatic License Plate Recognition in Unconstrained Scenarios," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 6 (2022): 5693–5703, https://doi.org/10.1109/TITS.2021.3055946.

19. Y.-Y. Liu, Q. Liu, S.-L. Chen, F. Chen, and X.-C. Yin, "Irregular License Plate Recognition via Global Information Integration," in *International Conference on Multimedia Modeling* (Springer, 2024), 325–339, https://doi.org/10.1007/978-3-031-53308-2_24.

20. R. Laroca, M. Santos, V. Estevam, E. Luz, and D. Menotti, "A First Look at Dataset Bias in License Plate Recognition," in *Conference on Graphics, Patterns and Images (SIBGRAPI)* (Oct 2022), 234–239, https://doi.org/10.1109/SIBGRAPI55357.2022.9991768.

21. A. Torralba and A. A. Efros, "Unbiased Look at Dataset Bias," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2011), 1521–1528, https://doi.org/10.1109/CVPR.2011.5995347.

22. T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Robust License Plate Recognition Using Neural Networks Trained on Synthetic Images," *Pattern Recognition* 93 (2019): 134–146, https://doi.org/10.1016/j.patcog.2019.04.007.

23. B.-G. Han, J. T. Lee, K.-T. Lim, and D.-H. Choi, "License Plate Image Generation Using Generative Adversarial Networks for End-to-End License Plate Character Recognition from a Small Set of Real Images," *Applied Sciences* 10, no. 8 (2020): 2780, https://doi.org/10.3390/app10082780.

24. Y. Gao, H. Lu, S. Mu, and S. Xu, "GroupPlate: Toward Multi-Category License Plate Recognition," *IEEE Transactions on Intelligent Transportation Systems* 24, no. 5 (2023): 5586–5599, https://doi.org/10.1109/TITS.2023.3244827.

25. L. Y. Chan, A. Zimmer, J. L. d. Silva, and T. Brandmeier, "European Union Dataset and Annotation Tool for Real Time Automatic License Plate Detection and Blurring," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)* (2020), 1–6, https://doi.org/10.1109/ITSC45102.2020.9294240.

26. X. Kong, K. Wang, M. Hou, X. Hao, G. Shen, X. Chen, and F. Xia, "A Federated Learning-Based License Plate Recognition Scheme for 5G-Enabled Internet of Vehicles," *IEEE Transactions on Industrial Informatics* 17, no. 12 (2021): 8523–8530, https://doi.org/10.1109/TII.2021.3067324.

27. L. Trinh, P. Pham, H. Trinh, N. Bach, D. Nguyen, G. Nguyen, and H. Nguyen, "PP4AV: A Benchmarking Dataset for Privacy-Preserving Autonomous Driving," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2023), 1206–1215, https://doi.org/10.1109/WACV56688.2023.00126.

28. A. Mecocci and C. Tommaso, "Generative Models for License Plate Recognition by Using a Limited Number of Training Samples," in *International Conference on Image Processing (ICIP)* (IEEE, 2006), 2769–2772, https://doi.org/10.1109/ICIP.2006.313121.

29. C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License Plate Recognition from Still Images and Video Sequences: A Survey," *IEEE Transactions on Intelligent Transportation Systems* 9, no. 3 (2008): 377–391, https://doi.org/10.1109/TITS.2008.922938.

30. L. Zhang, P. Wang, H. Li, Z. Li, C. Shen, and Y. Zhang, "A Robust Attentional Framework for License Plate Recognition in the Wild," *IEEE Transactions on Intelligent Transportation Systems* 22, no. 11 (2021): 6967–6976, https://doi.org/10.1109/TITS.2020.3000072.

31. T. Wang, W. Wang, C. Li, and J. Tang, "Efficient License Plate Recognition via Parallel Position-Aware Attention," in *Pattern Recognition and Computer Vision* (IEEE, 2022), 346–360, https://doi.org/10.1007/978-3-031-18913-5_27.

32. N. Shvai, A. Hasnat, and A. Nakib, "Multiple Auxiliary Classifiers GAN for Controllable Image Generation: Application to License Plate Recognition," *IET Intelligent Transport Systems* 17, no. 1 (2023): 243–254, https://doi.org/10.1049/itr2.12251.

33. A. Maier, D. Moussa, A. Spruck, J. Seiler, and C. Riess, "Reliability Scoring for the Recognition of Degraded License Plates," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2022), 1–8, https://doi.org/10.1109/AVSS56176.2022.9959390.

34. S. Wu, W. Zhai, and Y. Cao, "PixTextGAN: Structure Aware Text Image Synthesis for License Plate Recognition," *IET Image Processing* 13, no. 14 (2019): 2744–2752, https://doi.org/10.1049/iet-ipr.2018.6588.

35. G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Real-Time Automatic License Plate Recognition through Deep Multi-Task Networks," in *Conference on Graphics, Patterns and Images (SIBGRAPI)* (IEEE, 2018), 110–117, https://doi.org/10.1109/SIBGRAPI.2018.00021.

36. J. Shashirangana, H. Padmasiri, D. Meedeniya, C. Perera, S. R. Nayak, J. Nayak, S. Vimal, and S. Kadry, "License Plate Recognition Using Neural Architecture Search for Edge Devices," *International Journal of Intelligent Systems* 37, no. 12 (2022): 10211–10248, https://doi.org/10.1002/int.22471.

37. R. Al-batat, A. Angelopoulou, S. Premkumar, J. Hemanth, and E. Kapetanios, "An End-to-End Automated License Plate Recognition System Using YOLO Based Vehicle and License Plate Detection with Vehicle Classification," *Sensors* 22, no. 23 (2022): 9477, https://doi.org/10.3390/s22239477.

38. R. Laroca, L. A. Zanlorensi, V. Estevam, R. Minetto, and D. Menotti, "Leveraging Model Fusion for Improved License Plate Recognition," in *Iberoamerican Congress on Pattern Recognition (CIARP)* (Springer, 2023), 60–75, https://doi.org/10.1007/978-3-031-49249-5_5.

39. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), 5967–5976, https://doi.org/10.1109/CVPR.2017.632.

40. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2017), 2242–2251, https://doi.org/10.1109/ICCV.2017.244.

41. R. Laroca, V. Barroso, M. A. Diniz, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, “Convolutional Neural Networks for automatic meter reading,” *Journal of Electronic Imaging* 28, no. 1 (2019): 013023, https://doi.org/10.1117/1.JEI.28.1.013023.

42. S. M. Silva and C. R. Jung, “Real-time license plate detection and recognition using deep convolutional neural networks,” *Journal of Visual Communication and Image Representation* (2020): 102773, https://doi.org/10.1016/j.jvcir.2020.102773.

43. Y. Zhang, Z. Wang, and J. Zhuang, “Efficient license plate recognition via holistic position attention,” in *AAAI Conference on Artificial Intelligence* (May 2021), 3438–3446, https://doi.org/10.1609/aaai.v35i4.16457.

44. Y. Wang, Z.-P. Bian, Y. Zhou, and L.-P. Chau, “Rethinking and designing a high-performing automatic license plate recognition approach,” *IEEE Transactions on Intelligent Transportation Systems* 23, no. 7 (2022): 8868–8880, https://doi.org/10.1109/TITS.2021.3087158.

45. M. Weber, “Caltech Cars Dataset,” 1999, https://data.caltech.edu/records/20084.

46. V. Srebrić, “EnglishLP Database,” 2003, https://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip.

47. L. Dlagnekov and S. Belongie, “UCSD/Calit2 Car License Plate, Make and Model Database,” 2005, https://www.belongielab.org/car_data.html.

48. W. Zhou, H. Li, Y. Lu, and Q. Tian, “Principal Visual Word Discovery for Automatic License Plate Detection,” *IEEE Transactions on Image Processing* 21, no. 9 (Sept 2012): 4269–4279, https://doi.org/10.1109/TIP.2012.2199506.

49. G. S. Hsu, J. C. Chen, and Y. Z. Chung, “Application-Oriented License Plate Recognition,” *IEEE Transactions on Vehicular Technology* 62, no. 2 (Feb 2013): 552–561, https://doi.org/10.1109/TVT.2012.2226218.

50. OpenALPR, “OpenALPR-EU Dataset,” 2016, https://github.com/openalpr/benchmarks/tree/master/endtoend/eu.

51. G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, “Benchmark for License Plate Character Segmentation,” *Journal of Electronic Imaging* 25, no. 5 (2016): 053034, https://doi.org/10.1117/1.JEI.25.5.053034.

52. Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, “A Robust and Efficient Approach to License Plate Detection,” *IEEE Transactions on Image Processing* 26, no. 3 (March 2017): 1102–1114, https://doi.org/10.1109/TIP.2016.2631901.

53. R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, “A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector,” in *International Joint Conference on Neural Networks (IJCNN)* (IEEE, July 2018), 1–10, https://doi.org/10.1109/IJCNN.2018.8489629.

54. S. M. Silva and C. R. Jung, “License Plate Detection and Recognition in Unconstrained Scenarios,” in *European Conference on Computer Vision (ECCV)* (Springer, 2018), 593–609, https://doi.org/10.1007/978-3-030-01258-8_36.

55. R. Laroca, E. V. Cardoso, D. R. Lucio, V. Estevam, and D. Menotti, “On the Cross-Dataset Generalization in License Plate Recognition,” in *International Conference on Computer Vision Theory and Applications (VISAPP)* (IEEE, 2022), 166–178, https://doi.org/10.5220/0010846800003124.

56. S.-L. Chen, Q. Liu, F. Chen, and X.-C. Yin, “End-to-End Multi-Line License Plate Recognition with Cascaded Perception,” in *International Conference on Document Analysis and Recognition (ICDAR)* (IEEE, 2023), 274–289, https://doi.org/10.1007/978-3-031-41734-4_17.

57. X. Zhang, N. Gu, H. Ye, and C. Lin, “Vehicle License Plate Detection and Recognition Using Deep Neural Networks and Generative Adversarial Networks,” *Journal of Electronic Imaging* 27, no. 4 (2018): 043056, https://doi.org/10.1117/1.JEI.27.4.043056.

58. H. Xiang, Y. Zhao, Y. Yuan, G. Zhang, and X. Hu, “Lightweight Fully Convolutional Network for License Plate Detection,” *Optik* 178 (2019): 1185–1194, https://doi.org/10.1016/j.ijleo.2018.10.098.

59. Q. Liu, Y. Liu, S.-L. Chen, T.-H. Zhang, F. Chen, and X.-C. Yin, “Improving Multi-Type License Plate Recognition via Learning Globally and Contrastively,” *IEEE Transactions on Intelligent Transportation Systems* 25, no. 9 (2024): 1–11, https://doi.org/10.1109/TITS.2024.3365537.

60. R. Panahi and I. Gholampour, “Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications,” *IEEE Transactions on Intelligent Transportation Systems* 18, no. 4 (April 2017): 767–779, https://doi.org/10.1109/TITS.2016.2586520.

61. M. S. Beratoğlu and B. U. Töreyiṅ, “Vehicle license plate detector in compressed domain,” *IEEE Access* 9 (2021): 95087–95096, https://doi.org/10.1109/ACCESS.2021.3092938.

62. J. Zhuang, S. Hou, Z. Wang, and Z.-J. Zha, “Towards Human-Level License Plate Recognition,” in *European Conference on Computer Vision (ECCV)* (Springer, 2018), 314–329, https://doi.org/10.1007/978-3-030-01219-9_19.

63. J. Liang, G. Chen, Y. Wang, and H. Qin, “EGSANet: Edge-Guided Sparse Attention Network for Improving License Plate Detection in the Wild,” *Applied Intelligence* 52, no. 4 (2022): 4458–4472, https://doi.org/10.1007/s10489-021-02628-4.

64. A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and Flexible Image Augmentations,” *Information* 11, no. 2 (2020): 125, https://doi.org/10.3390/info11020125.

65. L. Qiao, Y. Chen, Z. Cheng, Y. Xu, Y. Niu, S. Pu, and F. Wu, “MANGO: A Mask Attention Guided One-Stage Scene Text Spotter,” *AAAI Conference on Artificial Intelligence* 35, no. 3 (May 2021): 2467–2476.

66. Y. Zou, Y. Zhang, J. Yan, X. Jiang, T. Huang, H. Fan, and Z. Cui, “A Robust License Plate Recognition Model Based on Bi-LSTM,” *IEEE Access* 8 (2020): 211630–211641, https://doi.org/10.1109/ACCESS.2020.3040238.

67. C. Lee and S. Osindero, “Recursive recurrent nets with attention modeling for OCR in the wild,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 2231–2239, https://doi.org/10.1109/CVPR.2016.245.

68. B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, “Robust Scene Text Recognition with Automatic Rectification,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 4168–4176, https://doi.org/10.1109/CVPR.2016.452.

69. W. Liu, C. Chen, Z. S. Kwan-Yee K. Wong, and J. Han, “STAR-Net: A Spatial Attention Residue Network for Scene Text Recognition,” in *British Machine Vision Conference (BMVC)* (The British Machine Vision Association, Sept 2016), 1–13, https://doi.org/10.5244/C.30.43.

70. B. Shi, X. Bai, and C. Yao, “An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 11 (Nov 2017): 2298–2304, https://doi.org/10.1109/TPAMI.2016.2646371.

71. J. Wang and X. Hu, “Gated Recurrent Convolution Neural Network for OCR,” in *International Conference on Neural Information Processing Systems (NeurIPS)* (Curran Associates, Inc., 2017), 334–343, https://doi.org/10.5555/3294771.3294803.

72. F. Borisyuk, A. Gordo, and V. Sivakumar, “Rosetta: Large Scale System for Text Detection and Recognition in Images,” in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, 2018), 71–79, https://doi.org/10.1145/3219819.3219861.

73. J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, “What is Wrong with Scene Text Recognition Model Comparisons? Dataset and Model Analysis,” in *IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2019), 4714–4722, https://doi.org/10.1109/ICCV.2019.00481.

74. R. Atienza, “Vision Transformer for Fast and Efficient Scene Text Recognition,” in *International Conference on Document Analysis and Recognition (ICDAR)* (IEEE, 2021), 319–334, https://doi.org/10.1007/978-3-030-86549-8_21.

75. J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík, “Holistic Recognition of Low Quality License Plates by CNN Using Track Annotated Data,” in *IEEE International Conference on Advanced Video and Signal Based Surveillance* (IEEE, Aug 2017), 1–6, https://doi.org/10.1109/AVSS.2017.8078501.

76. G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, “Multi-Task Learning for Low-Resolution License Plate Recognition,” in *Iberoamerican Congress on Pattern Recognition (CIARP)* (Oct 2019): 251–261, https://doi.org/10.1007/978-3-030-33904-3_23.

77. R. Laroca, A. B. Araujo, L. A. Zanlorensi, E. C. De Almeida, and D. Menotti, “Towards Image-Based Automatic Meter Reading in Unconstrained Scenarios: A robust and efficient approach,” *IEEE Access* 9 (2021): 67 569–67 584, https://doi.org/10.1109/ACCESS.2021.3077415.

78. J. Baek, Y. Matsui, and K. Aizawa, “What If We Only Use Real Datasets for Scene Text Recognition? Toward Scene Text Recognition with Fewer Labels,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021), 3112–3121, https://doi.org/10.1109/CVPR46437.2021.00313.

79. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Scaled-YOLOv4: Scaling Cross Stage Partial Network,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, June 2021), 13 029–13 038, https://doi.org/10.1109/CVPR46437.2021.01283.

80. H. Xu, X.-D. Zhou, Z. Li, L. Liu, C. Li, and Y. Shi, “EILPR: Toward End-to-End Irregular License Plate Recognition Based on Automatic Perspective Alignment,” *IEEE Transactions on Intelligent Transportation Systems* 23, no. 3 (2022): 2586–2595, https://doi.org/10.1109/TITS.2021.3130898.

81. Y. Jiang, F. Jiang, H. Luo, H. Lin, J. Yao, J. Liu, and J. Ren, “An Efficient and Unified Recognition Method for Multiple License Plates in Unconstrained Scenarios,” *IEEE Transactions on Intelligent Transportation Systems* 24, no. 5 (2023): 5376–5389, https://doi.org/10.1109/TITS.2023.3237743.

82. H. Li, P. Wang, and C. Shen, “Toward End-to-End Car License Plate Detection and Recognition with Deep Neural Networks,” *IEEE Transactions on Intelligent Transportation Systems* 20, no. 3 (2019): 1126–1136, https://doi.org/10.1109/TITS.2018.2847291.

83. W. Jia and M. Xie, “An Efficient License Plate Detection Approach With Deep Convolutional Neural Networks in Unconstrained Scenarios,” *IEEE Access* 11 (2023): 85626–85639, https://doi.org/10.1109/ACCESS.2023.3301122.

84. A. Meng, W. Yang, Z. Xu, H. Huang, L. Huang, and C. Ying, “A Robust and Efficient Method for License Plate Recognition,” in *International Conference on Pattern Recognition (ICPR)* (IEEE, 2018), 1713–1718, https://doi.org/10.1109/ICPR.2018.8546291.

85. H. Yoo and K. Jun, “Deep Corner Prediction to Rectify Tilted License Plate Images,” *Multimedia Systems* 27 (2021): 779–786, https://doi.org/10.1007/s00530-020-00655-8.

86. C. Zhang, Q. Wang, and X. Li, “V-LPDR: Towards a Unified Framework for License Plate Detection, Tracking, and Recognition in Real-World Traffic Videos,” *Neurocomputing* 449 (2021): 189–206, https://doi.org/10.1016/j.neucom.2021.03.103.

87. X. Zhou, Y. Cheng, L. Jiang, and B. Ning, “FAFEnet: A Fast and Accurate Model for Automatic License Plate Detection and Recognition,” *IET Image Processing* 17 (2023): 807–818, https://doi.org/10.1049/ipr2.12674.

88. Sighthound, “Sighthound Cloud API,” 2023, https://www.sighthound.com/products/cloud.

89. OpenALPR, “OpenALPR Cloud API,” http://www.openalpr.com/.

90. R. D. Castro-Zunti, J. Yépez, and S.-B. Ko, “License Plate Segmentation and Recognition System Using Deep Learning and OpenVINO,” *IET Intelligent Transport Systems* 14 (2020): 119–126, https://doi.org/10.1049/iet-its.2019.0481.

91. Y. Zou, Y. Zhang, J. Yan, et al., “License Plate Detection and Recognition Based on YOLOv3 and ILPRNET,” *Signal, Image and Video Processing* 16 (2022): 473–480, https://doi.org/10.1007/s11760-021-01981-8.

92. R. Laroca, V. Estevam, A. S. Britto Jr., R. Minetto, and D. Menotti, “Do We Train on Test Data? The Impact of Near-Duplicates on License Plate Recognition,” in *International Joint Conference on Neural Networks (IJCNN)* (IEEE, Jun 2023), 1–8, https://doi.org/10.1109/IJCNN54540.2023.10191584.