






# Leveraging Model Fusion for Improved License Plate Recognition<sup>\*</sup>

Rayson Laroca<sup>1</sup> , Luiz A. Zanlorensi<sup>1</sup> , Valter Estevam<sup>1,2</sup> ,  
Rodrigo Minetto<sup>3</sup> , and David Menotti<sup>1</sup> 

<sup>1</sup> Federal University of Paraná, Curitiba, Brazil

<sup>2</sup> Federal Institute of Paraná, Irati, Brazil

<sup>3</sup> Federal University of Technology-Paraná, Curitiba, Brazil

**Abstract.** License Plate Recognition (LPR) plays a critical role in various applications, such as toll collection, parking management, and traffic law enforcement. Although LPR has witnessed significant advancements through the development of deep learning, there has been a noticeable lack of studies exploring the potential improvements in results by fusing the outputs from multiple recognition models. This research aims to fill this gap by investigating the combination of up to 12 different models using straightforward approaches, such as selecting the most confident prediction and employing majority vote-based strategies. Our experiments encompass a wide range of datasets, revealing substantial benefits of fusion approaches in both intra- and cross-dataset setups. Essentially, fusing multiple models reduces considerably the likelihood of obtaining subpar performance on a particular dataset/scenario. We also found that combining models based on their speed is a compelling approach. Specifically, for applications where the recognition task can tolerate some additional time, though not excessively, an effective strategy is to combine 4–6 fast models. These models may not be the most accurate individually, but their fusion strikes an optimal balance between accuracy and speed.

**Keywords:** License Plate Recognition · Model Fusion · Ensemble.

## 1 Introduction

Automatic License Plate Recognition (ALPR) has garnered substantial interest in recent years due to its many practical applications, which include toll collection, parking management, border control, and road traffic monitoring [18, 22, 42].

In the deep learning era, ALPR systems customarily comprise two key components: license plate detection (LPD) and license plate recognition (LPR). LPD entails locating regions containing license plates (LPs) within an image, while LPR involves identifying the characters within these LPs. Recent research has predominantly concentrated on advancing LPR [25, 28, 46], given that widely

---

<sup>\*</sup> Supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) (# 88881.516265/2020-01), and by the National Council for Scientific and Technological Development (CNPq) (# 309953/2019-7 and # 308879/2020-1).

adopted object detectors such as Faster-RCNN and YOLO have consistently delivered impressive results in LPD for some years now [14, 21, 47].

This study also focuses on LPR but provides a unique perspective compared to recent research. Although deep learning techniques have enabled significant advancements in this field over the past years, multiple studies have shown that different models exhibit varying levels of robustness across different datasets [20, 27, 45]. Each dataset poses distinct challenges, such as diverse LP layouts and varying tilt ranges. As a result, a method that performs optimally in one dataset may yield poor results in another. This raises an important question: *“Can we substantially enhance LPR results by fusing the outputs of diverse recognition models?”* If so, two additional questions arise: *“To what extent can this improvement be attained?”* and *“How many and which models should be employed?”* As of now, such questions remain unanswered in the existing literature.

We acknowledge that some ALPR applications impose stringent time constraints on their execution. This is particularly true for embedded systems engaged in tasks such as access control and parking management in high-traffic areas. However, in other contexts, such as systems used for issuing traffic tickets and conducting forensic investigations, there is often a preference to prioritize the recognition rate, even if it sacrifices efficiency [16, 28, 31]. These scenarios can greatly benefit from the fusion of multiple recognition models.

Taking this into account, in this study, we thoroughly examine the potential of enhancing LPR results through the fusion of outputs from multiple recognition models. Remarkably, we assess the combination of up to 12 well-known models across 12 different datasets, setting our investigation apart from earlier studies.

In summary, this paper has two main contributions:

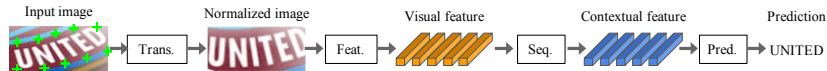
- We present compelling evidence showcasing the benefits offered by fusion approaches in both intra- and cross-dataset setups. In the intra-dataset setup, the mean recognition rate across the datasets experiences a substantial boost, rising from 92.4% achieved by the best model individually to 97.6% when leveraging the best fusion approach. Similarly, in the cross-dataset setup, the mean recognition rate increases from 87.6% to levels exceeding 90%. Notably, in both setups, the sequence-level majority vote fusion approach outperform both character-level majority vote and selecting the prediction made with the highest confidence approaches.
- We draw attention to the effectiveness of fusing models based on their speed. This approach is particularly useful for applications where the recognition task can accommodate a moderate increase in processing time. In such cases, the recommended strategy is to combine 4–6 fast models. Although these models may not achieve the highest accuracy individually, their fusion results in an optimal trade-off between accuracy and speed.

The rest of this paper is structured as follows. Section 2 provides a concise overview of the recognition models explored in this work. The experimental setup adopted in our research is detailed in Section 3. The results obtained are presented and analyzed in Section 4. Lastly, Section 5 concludes the paper by summarizing the findings and suggesting potential avenues for future research.

## 2 Related Work

LPR is widely recognized as a specific application within the field of scene text recognition [7, 24, 49]. LPR sets itself apart primarily due to the limited presence of strong linguistic context information and the minimal variation observed between characters. The following paragraphs briefly describe well-known models originally proposed for general scene text recognition, LPR, and related tasks. These models will be explored in this study.

Baek et al. [2] introduced a four-stage framework (depicted in Fig. 1) that models the design patterns of most modern methods for scene text recognition. The *Transformation* stage removes the distortion from the input image so that the text is horizontal or normalized. This task is generally done through spatial transformer networks with a thin-plate splines (TPS) transformation, which models the distortion by finding and correcting fiducial points. The second stage, *Feature Extraction*, maps the input image to a representation that focuses on the attributes relevant to character recognition while suppressing irrelevant features such as font, color, size and background. This task is usually performed by a module composed of Convolutional Neural Networks (CNNs), such as VGG, ResNet, and RCNN. The *Sequence Modeling* stage converts visual features to contextual features that capture the context in the sequence of characters. Bi-directional Long Short-Term Memory (Bi-LSTM) is generally employed for this task. Finally, the *Prediction* stage produces the character sequence from the identified features. This task is typically done by a Connectionist Temporal Classification (CTC) decoder or through an attention mechanism. As can be seen in Table 1, while most methods can fit within this framework, they do not necessarily incorporate all four modules.



**Fig. 1.** The four modules or stages of modern scene text recognition, according to [2]. “Trans.” stands for Transformation, “Feat.” stands for Feature Extraction, “Seq.” stands for Sequence Modeling, and “Pred.” stands for Prediction. Image reproduced from [2].

**Table 1.** Summary of seven well-known models for scene text recognition. These models are listed chronologically and are further explored in other sections of this work.

Model	Transformation	Feature Extraction	Sequence Modeling	Prediction
R <sup>2</sup> AM [23]	–	RCNN	–	Attention
RARE [33]	TPS	VGG	Bi-LSTM	Attention
STAR-Net [26]	TPS	ResNet	Bi-LSTM	CTC
CRNN [32]	–	VGG	Bi-LSTM	CTC
GRCNN [41]	–	RCNN	Bi-LSTM	CTC
Rosetta [4]	–	ResNet	–	CTC
TRBA [2]	TPS	ResNet	Bi-LSTM	Attention

Atienza [1] drew inspiration from the accomplishments of the Vision Transformer (ViT) and put forward a single-stage model named ViTSTR for scene text recognition. It operates by initially dividing the input image into non-overlapping patches. These patches are then converted into 1-D vector embeddings (i.e., flattened 2-D patches). To feed the encoder, each embedding is supplemented with a learnable patch embedding and a corresponding position encoding.

Recent works on LPR have focused on developing multi-task CNNs that can process the entire LP image holistically, eliminating the need for character segmentation [6, 11, 38]. Two such models are Holistic-CNN [38] and Multi-task [11]. In these models, the LP image undergoes initial processing via convolutional layers, followed by  $N$  branches of fully connected layers. Each branch is responsible for predicting a single character class (including a ‘blank’ character) at a specific position on the LP, enabling the branches to collectively predict up to  $N$  characters. Both models are often used as baselines due to their remarkable balance between speed and accuracy [12, 20, 25, 28, 37].

The great speed/accuracy trade-off provided by YOLO networks [40] has inspired many authors to explore similar architectures targeting real-time performance for LPR and similar tasks. Silva & Jung [35] proposed CR-NET, a YOLO-based model that effectively detects and recognizes all characters within a cropped LP. CR-NET incorporates the initial eleven layers of YOLOv1, along with four additional convolutional layers to enhance nonlinearity. Recent studies [20, 22, 36] have reported impressive results using CR-NET. Another noteworthy model is Fast-OCR [19], which incorporates features from several object detectors that prioritize the trade-off between speed and accuracy. To illustrate, Fast-OCR performs detection at two different scales, as Tiny-YOLOv4, and its convolutional layers mostly have  $3 \times 3$  kernels, with the number of filters being doubled after each max-pooling layer, as in YOLOv2 and CR-NET. In [19], Fast-OCR achieved considerably better results than multiple baselines that perform recognition holistically, including CRNN [32], Multi-task [11] and TRBA [2].

While we found a few works leveraging model fusion to improve LPR results, we observed that they explored a limited range of models and datasets in the experiments. For example, Izidio et al. [16] employed multiple instances of the same model (i.e., Tiny-YOLOv3) rather than different models with varying architectures. Their experiments were conducted exclusively on a private dataset. Another example is the very recent work by Schirmacher et al. [31], where they examined deep ensembles, BatchEnsemble, and Monte Carlo dropout using multiple instances of two backbone architectures. The authors’ primary focus was on recognizing severely degraded images, leading them to perform nearly all of their experiments on a synthetic dataset containing artificially degraded images.

In summary, although the field of LPR has witnessed significant advancements through the development and application of deep learning-based models, there has been a noticeable lack of studies thoroughly examining the potential improvements in results by fusing the outputs from multiple recognition models.

### 3 Experimental Setup

This section provides an overview of the setup adopted in our experiments. We first enumerate the recognition models implemented for this study, providing specific information about the framework used for training and testing each of them, as well as the corresponding hyperparameters. Subsequently, we compile a list of the datasets employed in our assessments, showcasing sample LP images from each dataset to highlight their diversity. Afterward, we elaborate on the strategies examined for fusing the outputs of the different models. Finally, we describe how the performance evaluation is carried out.

The experiments were conducted on a PC with an AMD Ryzen Threadripper 1920X 3.5GHz CPU, 96 GB of RAM operating at 2133 MHz, an SSD (read: 535 MB/s; write: 445 MB/s), and an NVIDIA Quadro RTX 8000 GPU (48 GB).

#### 3.1 Recognition Models

We explore 12 recognition models in our experiments: RARE [33], R<sup>2</sup>AM [23], STAR-Net [26], CRNN [32], GRCNN [41], Holistic-CNN [38], Multi-task [11], Rosetta [4], TRBA [2], CR-NET [35], Fast-OCR [19] and ViTSTR-Base [1]. As discussed in Section 2, these models were chosen because they rely on design patterns shared by many renowned models for scene text recognition, as well as for their frequent roles as baselines in recent LPR research [12, 17, 20].

We implemented each model using the original framework or well-known public repositories associated with it. Specifically, we employed Darknet<sup>1</sup> for implementing the YOLO-based models (CR-NET and Fast-OCR). The multi-task models, Holistic-CNN and Multi-task, were trained and evaluated using Keras. As for the remaining models, which were originally proposed for general scene text recognition, we used a fork<sup>2</sup> of the open source repository of Clova AI Research (PyTorch). This repository has gained recognition for reaching first place in the ICDAR2013 focused scene text and ICDAR2019 ArT competitions, as well as third place in ICDAR2017 COCO-Text and ICDAR2019 ReCTS (task1) [2].

Here we list the hyperparameters employed in each framework for training the recognition models. These hyperparameters were determined based on existing research [1, 2, 20] and were further validated through experiments on the validation set. In Darknet, the parameters include: Stochastic Gradient Descent (SGD) optimizer, 90K iterations, a batch size of 64, and a learning rate of  $[10^{-3}, 10^{-4}, 10^{-5}]$  with decay steps at 30K and 60K iterations. In Keras, we employed the Adam optimizer with an initial learning rate of  $10^{-3}$  (ReduceLROnPlateau’s patience of 5 and factor of  $10^{-1}$ ), a batch size of 64, and a patience value of 11 (patience indicates the number of epochs without improvement before training is stopped). In PyTorch, we used the following parameters: Adadelta optimizer with a decay rate of  $\rho = 0.99$ , 300K iterations, and a batch size of 128. The only modification we made to the models’ architectures was adjusting the respective input layers to accommodate images with a width-to-height ratio of 3.

<sup>1</sup> <https://github.com/AlexeyAB/darknet>

<sup>2</sup> <https://github.com/roattienza/deep-text-recognition-benchmark/>

### 3.2 Datasets

Researchers have conducted experiments on various datasets to showcase the effectiveness of their systems in recognizing LPs from different regions [13, 22, 24, 34]. As shown in Table 2, we perform experiments using images from 12 public datasets commonly used to benchmark ALPR systems [18, 20, 36, 45]. Each dataset was divided using standard splits, defined by the datasets’ authors, or following previous works [18, 22, 42] (when there is no standard split)<sup>3</sup>. Specifically, eight datasets were used both for training and evaluating the recognition models, while four were used exclusively for testing. The selected datasets exhibit substantial diversity in terms of image quantity, acquisition settings, image resolution, and LP layouts. As far as we know, no other work in ALPR research has conducted experiments using images from such a wide range of public datasets.

**Table 2.** The datasets employed in our experimental analysis, with ‘\*’ indicating those used exclusively for testing (i.e., in cross-dataset experiments). The “Chinese” layout denotes LPs assigned to vehicles registered in mainland China, while the “Taiwanese” layout corresponds to LPs issued for vehicles registered in the Taiwan region.

Dataset	Year	Images	LP Layout	Dataset	Year	Images	LP Layout
Caltech Cars [43]	1999	126	American	SSIG-SegPlate [9]	2016	2,000	Brazilian
EnglishLP [39]	2003	509	European	PKU* [44]	2017	2,253	Chinese
UCSD-Stills [5]	2005	291	American	UFPR-ALPR [21]	2018	4,500	Brazilian
ChineseLP [48]	2012	411	Chinese	CD-HARD* [34]	2018	104	Various
AOLP [15]	2013	2,049	Taiwanese	CLPD* [46]	2021	1,200	Chinese
OpenALPR-EU* [29]	2016	108	European	RodoSol-ALPR [20]	2022	20,000	Brazilian & Mercosur

The diversity of LP layouts across the selected datasets is depicted in Fig. 2, revealing considerable variations even among LPs from the same region. For instance, the EnglishLP and OpenALPR-EU datasets, both collected in Europe, include images of LPs with notable distinctions in colors, aspect ratios, symbols (e.g., coats of arms), and the number of characters. Furthermore, certain datasets encompass LPs with two rows of characters, shadows, tilted orientations, and at relatively low spatial resolutions.

We explored various data augmentation techniques to ensure a balanced distribution of training images across different datasets. These techniques include random cropping, the introduction of random shadows, grayscale conversion, and random perturbations of hue, saturation, and brightness. Additionally, to counteract the propensity of recognition models to memorize sequence patterns encountered during training [8, 10, 45], we generated many synthetic LP images by shuffling the character positions on each LP (using the labels provided in [22]). Examples of these generated images are shown in Fig. 3.

<sup>3</sup> Detailed information on which images were used to train, validate and test the models can be accessed at <https://raysonlaroca.github.io/supp/lpr-model-fusion/>



**Fig. 2.** Some LP images from the public datasets used in our experimental evaluation.



**Fig. 3.** Examples of LP images we created to mitigate overfitting. Within each group, the image on the left is the original, while the remaining ones are artificially generated counterparts. Various transformations were applied to enhance image variability.

### 3.3 Fusion Approaches

This study examines three primary approaches to combine the outputs of multiple recognition models. The first approach involves selecting the sequence predicted with the *Highest Confidence* (HC) value as the final prediction, even if only one model predicts it. The second approach employs the *Majority Vote* (MV) rule to aggregate the sequences predicted by the different models. In other words, the final prediction is based on the sequence predicted by the largest number of models, disregarding the confidence values associated with each prediction. Lastly, the third approach follows a similar *Majority Vote* rule but performs individual aggregation for each *Character Position* (MVCP). To illustrate, the characters predicted in the first position are analyzed separately, and the character predicted the most times is selected. The same process is then applied to each subsequent character position until the last one. Ultimately, the selected characters are concatenated to form the final string.

One concern that arises when employing majority vote-based strategies is the potential occurrence of a tie. Let’s consider a scenario where an LP image is processed by five recognition models. Two models predict “ABC-123,” two models predict “ABC-124,” and the remaining model predicts “ABC-125.” In this case, a tie occurs between “ABC-123” and “ABC-124.” To address this, we assess two tie-breaking approaches for each majority vote strategy: (i) selecting the predic-



tion made with higher confidence among the tied predictions as the correct one, and (ii) selecting the prediction made by the “best model” as the correct one. In this study, for simplicity, we consider the best model the one that performs best individually across all datasets. However, in a more practical scenario, the chosen model could be the one known to perform best in the specific implementation scenario (e.g., one model may be the most robust for recognizing tilted LPs while another model may excel at handling low-resolution or noisy images). We intuitively use the acronym MV–HC to refer to the majority vote approach in which ties are broken by selecting the prediction made with the highest confidence value. Similarly, MV–BM refers to the majority vote approach in which ties are resolved by choosing the prediction made by the best model. The MVCP approach follows a similar naming convention.

It is important to mention that when conducting fusion based on the highest confidence, we consider the confidence values derived directly from the models’ outputs, even though some of them tend to make overconfident predictions. We carried out several experiments in which we normalized the confidence values of different models before fusing them, using various strategies such as weighted normalization based on the average confidence of each classifier’s predictions. Somewhat surprisingly, these attempts did not yield improved results.

### 3.4 Performance Evaluation

In line with the standard practice in the literature, we report the performance of each experiment by calculating the ratio of correctly recognized LPs to the total number of LPs in the test set. An LP is considered correctly recognized only if all the characters on the LP are accurately identified, as even a single incorrectly recognized character can lead to the misidentification of the vehicle.

It is important to note that, although this work focuses on the LPR stage, the LP images used as input for the recognition models were not directly cropped from the ground truth. Instead, the YOLOv4 model [3] was employed to detect the LPs. This approach allows for a more accurate simulation of real-world scenarios, considering the imperfect nature of LP detection and the reduced robustness of certain recognition models when faced with imprecisely detected LP regions [10, 24]. As in [20], the results obtained using YOLOv4 were highly satisfactory. Considering detections with an Intersection over Union (IoU)  $\geq 0.7$  as correct, YOLOv4 achieved an average recall rate exceeding 99.7% in the test sets of the datasets used for training and validation, and 98.0% in the cross-dataset experiments. In both cases, the precision rates obtained were around 98%.

## 4 Results

Table 3 shows the recognition rates obtained on the disjoint test sets of the eight datasets used for training and validating the models. It presents the results reached by each model individually, as well as the outcomes achieved through



the fusion strategies outlined in Section 3.3. To improve clarity, Table 3 only includes the best results attained through model fusion. For a detailed breakdown of the results achieved by combining the outputs from the top 2 to the top 12 recognition models, refer to Table 4. The ranking of the models was determined based on their mean performance across the datasets (the ranking on the validation set was essentially the same, with only two models swapping positions).

**Table 3.** Comparison of the recognition rates achieved across eight popular datasets by 12 models individually and through five different fusion strategies. Each model (rows) was trained once on the combined set of training images from all datasets and evaluated on the respective test sets (columns). The models are listed alphabetically, and the best recognition rates achieved in each dataset are shown in bold.

Test set \ Approach	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 159	AOLP # 683	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
CR-NET [35]	<b>97.8%</b>	94.1%	<b>100.0%</b>	<b>97.5%</b>	98.1%	<b>97.5%</b>	82.6%	59.0% <sup>†</sup>	90.8%
CRNN [32]	93.5%	88.2%	91.7%	90.7%	97.1%	92.9%	68.9%	73.6%	87.1%
Fast-OCR [19]	93.5%	<b>97.1%</b>	<b>100.0%</b>	<b>97.5%</b>	98.1%	97.1%	81.6%	56.7% <sup>†</sup>	90.2%
GRCNN [41]	93.5%	92.2%	93.3%	91.9%	97.1%	93.4%	66.6%	77.6%	88.2%
Holistic-CNN [38]	87.0%	75.5%	88.3%	95.0%	97.7%	95.6%	81.2%	94.7%	89.4%
Multi-task [11]	89.1%	73.5%	85.0%	92.5%	94.9%	93.3%	72.3%	86.6%	85.9%
R <sup>2</sup> AM [23]	89.1%	83.3%	86.7%	91.9%	96.5%	92.0%	75.9%	83.4%	87.4%
RARE [33]	95.7%	94.1%	95.0%	94.4%	97.7%	94.0%	75.7%	78.7%	90.7%
Rosetta [4]	89.1%	82.4%	93.3%	93.8%	97.5%	94.4%	75.5%	89.0%	89.4%
STAR-Net [26]	95.7%	96.1%	95.0%	95.7%	97.8%	96.1%	78.8%	82.3%	92.2%
TRBA [2]	93.5%	91.2%	91.7%	93.8%	97.2%	97.3%	83.4%	80.6%	91.1%
ViTSTR-Base [1]	87.0%	88.2%	86.7%	96.9%	<b>99.4%</b>	95.8%	<b>89.7%</b>	<b>95.6%</b>	<b>92.4%</b>
=====									
Fusion HC (top 6)	<b>97.8%</b>	95.1%	96.7%	<b>98.1%</b>	99.0%	96.6%	90.9%	93.5%	96.0%
Fusion MV-BM (top 8)	<b>97.8%</b>	<b>97.1%</b>	<b>100.0%</b>	<b>98.1%</b>	<b>99.7%</b>	98.4%	92.7%	96.4%	97.5%
Fusion MV-HC (top 8)	<b>97.8%</b>	<b>97.1%</b>	<b>100.0%</b>	<b>98.1%</b>	<b>99.7%</b>	99.1%	92.3%	<b>96.5%</b>	<b>97.6%</b>
Fusion MVCP-BM (top 9)	95.7%	96.1%	<b>100.0%</b>	<b>98.1%</b>	99.6%	99.0%	<b>92.8%</b>	96.4%	97.2%
Fusion MVCP-HC (top 9)	<b>97.8%</b>	96.1%	<b>100.0%</b>	<b>98.1%</b>	99.6%	<b>99.3%</b>	92.5%	96.3%	97.5%

<sup>†</sup>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.

**Table 4.** Average results obtained across the datasets by combining the output of the top  $N$  recognition models, ranked by accuracy, using five distinct strategies.

Models	HC	MV-BM	MV-HC	MVCP-BM	MVCP-HC
Top 1 (ViTSTR-Base)	92.4%	92.4%	92.4%	92.4%	92.4%
Top 2 (+ STAR-Net)	94.1%	92.4%	94.1%	92.4%	94.1%
Top 3 (+ TRBA)	94.2%	94.6%	94.9%	94.2%	94.2%
Top 4 (+ CR-NET)	95.2%	95.9%	96.3%	94.8%	95.9%
Top 5 (+ RARE)	95.5%	96.1%	96.6%	96.1%	96.2%
Top 6 (+ Fast-OCR)	<b>96.0%</b>	97.1%	97.0%	96.7%	96.9%
Top 7 (+ Rosetta)	95.4%	97.3%	97.2%	97.1%	97.0%
Top 8 (+ Holistic-CNN)	95.7%	<b>97.5%</b>	<b>97.6%</b>	96.1%	97.2%
Top 9 (+ GRCNN)	95.7%	97.5%	97.5%	<b>97.2%</b>	<b>97.5%</b>
Top 10 (+ R <sup>2</sup> AM)	95.5%	97.4%	97.2%	96.1%	96.6%
Top 11 (+ CRNN)	95.2%	97.1%	97.0%	96.5%	96.5%
Top 12 (+ Multi-task)	95.0%	97.0%	97.0%	95.5%	96.5%

Upon analyzing the results presented in Table 3, it becomes evident that model fusion has yielded substantial improvements. Specifically, the highest average recognition rate increased from 92.4% (ViTSTR-Base) to 97.6% by combining the outputs of multiple recognition models (MV-HC). While each model

individually obtained recognition rates below 90% for at least one dataset (three on average), all fusion strategies surpassed the 90% threshold across all datasets. Remarkably, in most cases, fusion led to recognition rates exceeding 95%.

The significance of conducting experiments on multiple datasets becomes apparent as we observe that the best overall model (ViTSTR-Base) did not achieve the top result in five of the eight datasets. Notably, ViTSTR-Base exhibited relatively poor performance on the Caltech Cars, EnglishLP, and UCSD-Stills datasets. We attribute this to two primary reasons: (i) these datasets are older, containing fewer training images, and this seems to impact certain models more than others (as explained in Section 3.2, we exploited data augmentation techniques to mitigate this issue); and (ii) these datasets were collected in the United States and Europe, regions known for having a higher degree of variability in LP layouts compared to the regions where the other datasets were collected (specifically, Brazil, mainland China, and Taiwan). It is worth noting that we included these datasets in our experimental setup, despite their limited number of images, precisely because they provide an opportunity to uncover such valuable insights.

Basically, by analyzing the results reported for each dataset individually, we observe that combining the outputs of multiple models does not necessarily lead to significantly improved performance compared to the best model in the ensemble. Instead, it reduces the likelihood of obtaining poor performance. This phenomenon arises because diverse models tend to make different errors for each sample, but generally concur on correct classifications [30]. Illustrated in Fig. 4 are representative examples of predictions made by multiple models and the MV-HC fusion strategy on various LP images. It is remarkable that model fusion can produce accurate predictions even in cases where most models exhibit prediction errors. To clarify, with the MV-HC approach, this occurs when each incorrect sequence is predicted fewer times than the correct one, or in the case of a tie, the correct sequence is predicted with higher confidence.

Shifting our attention back to Table 4, we note that the majority vote-based strategies yielded comparable results, with the sequence-level approach (MV) performing marginally better for a given number of combined models. Our analysis indicates that this difference arises in cases where a model predicts one character more or one character less, impacting the majority vote by character position (MVCP) approach relatively more. Conversely, selecting the prediction with the highest confidence (HC) consistently led to inferior results. This can be attributed to the general tendency of all models to make incorrect predictions also with high confidence, as demonstrated in Fig. 4.

A growing number of authors [20, 42, 45] have stressed the importance of also evaluating LPR models in a cross-dataset fashion, as it more accurately simulates real-world scenarios where new cameras are regularly being deployed in new locations without existing systems being retrained as often. Taking this into account, we present in Table 5 the results obtained on four distinct datasets, none of which were used during the training of the models. These particular datasets are commonly employed for this purpose [6, 18, 22, 36, 49].



**Fig. 4.** Predictions obtained in eight LP images by multiple models individually and through the best fusion approach. Although we only show the predictions from the top 5 models for better viewing, it is noteworthy that in these particular cases, fusing the top 8 models (the optimal configuration) yielded identical predictions. The confidence for each prediction is indicated in parentheses, and any errors are highlighted in red.

**Table 5.** Comparison of the results achieved in cross-dataset setups by 12 models individually and through five different fusion strategies. The models are listed alphabetically, with the highest recognition rates attained for each dataset highlighted in bold.

Approach \ Dataset	OpenALPR-EU # 108	PKU # 2,253	CD-HARD # 104	CLPD # 1,200	Average
CR-NET [35]	96.3%	99.1%	58.7%	94.2%	87.1%
CRNN [32]	93.5%	98.2%	31.7%	89.0%	78.1%
Fast-OCR [19]	<b>97.2%</b>	<b>99.3%</b>	<b>59.6%</b>	<b>94.4%</b>	<b>87.6%</b>
GRCNN [41]	87.0%	98.6%	38.5%	87.7%	77.9%
Holistic-CNN [38]	89.8%	98.6%	11.5%	90.2%	72.5%
Multi-task [11]	85.2%	97.4%	10.6%	86.8%	70.0%
R <sup>2</sup> AM [23]	88.9%	97.1%	20.2%	88.2%	73.6%
RARE [33]	94.4%	98.3%	37.5%	92.4%	80.7%
Rosetta [4]	90.7%	97.2%	14.4%	86.9%	72.3%
STAR-Net [26]	<b>97.2%</b>	99.2%	48.1%	93.3%	84.4%
TRBA [2]	93.5%	98.5%	35.6%	90.9%	79.6%
ViTSTR-Base [1]	89.8%	98.4%	22.1%	93.1%	75.9%
=====					
Fusion HC ( <i>top 6</i> )	95.4%	99.2%	48.1%	94.9%	84.4%
Fusion MV-BM ( <i>top 8</i> )	<b>99.1%</b>	<b>99.7%</b>	<b>65.4%</b>	<b>97.0%</b>	<b>90.3%</b>
Fusion MV-HC ( <i>top 8</i> )	<b>99.1%</b>	<b>99.7%</b>	<b>65.4%</b>	96.3%	90.1%
Fusion MVCP-BM ( <i>top 9</i> )	95.4%	<b>99.7%</b>	54.8%	95.5%	86.3%
Fusion MVCP-HC ( <i>top 9</i> )	97.2%	<b>99.7%</b>	57.7%	95.9%	87.6%

These experiments provide further support for the findings presented earlier in this section. Specifically, both strategies that rely on a majority vote at the sequence level (MV-BM and MV-HC) outperformed the others significantly. The most notable performance disparity was observed in the CD-HARD dataset, known for its challenges due to a predominance of heavily tilted LPs (as shown in Fig. 2). Interestingly, in this scenario, the MV-BM approach exhibited slightly superior performance compared to MV-HC. What surprised us the most was

that the HC approach failed to yield any improvements in results on any dataset, indicating that the models made errors with high confidence even on LP images extracted from datasets that were not part of their training.

While our primary focus lies on investigating the improvements in recognition rates achieved through model fusion, it is also pertinent to examine its impact on runtime. Naturally, certain applications might favor combining fewer models to attain a moderate improvement in recognition while minimizing the increase in the system’s running time. With this in mind, Table 6 presents the number of frames per second (FPS) processed by each model independently and when incorporated into the ensemble. In addition to combining the models based on their average recognition rate across the datasets, as done in the rest of this section, we also explore combining them based on their processing speed.

**Table 6.** The number of FPS processed by each model independently and when incorporated into the ensembles. On the left, the models are ranked based on their results across the datasets, while on the right they are ranked according to their speed. The reported time, measured in milliseconds per image, represents the average of 5 runs.

Models (ranked by <b>accuracy</b> )	MV-HC	Individual		Fusion	
		Time	FPS	Time	FPS
Top 1 (ViTSTR-Base)	92.4%	7.3	137	7.3	137
Top 2 (+ STAR-Net)	94.1%	7.1	141	14.4	70
Top 3 (+ TRBA)	94.9%	16.9	59	31.3	32
Top 4 (+ CR-NET)	96.3%	5.3	189	36.6	27
Top 5 (+ RARE)	96.6%	13.0	77	49.6	20
Top 6 (+ Fast-OCR)	97.0%	3.0	330	52.6	19
Top 7 (+ Rosetta)	97.2%	4.6	219	57.2	18
Top 8 (+ Holistic-CNN)	97.6%	2.5	399	59.7	17
Top 9 (+ GRCNN)	97.5%	8.5	117	68.2	15
Top 10 (+ R <sup>2</sup> AM)	97.2%	15.9	63	84.2	12
Top 11 (+ CRNN)	97.0%	2.9	343	87.1	11
Top 12 (+ Multi-task)	97.0%	2.3	427	89.4	11

Models (ranked by <b>speed</b> )	MV-HC	Individual		Fusion	
		Time	FPS	Time	FPS
Top 1 (Multi-task)	85.9%	2.3	427	2.3	427
Top 2 (+ Holistic-CNN)	90.2%	2.5	399	4.9	206
Top 3 (+ CRNN)	91.1%	2.9	343	7.8	129
Top 4 (+ Fast-OCR)	95.4%	3.0	330	10.8	93
Top 5 (+ Rosetta)	96.0%	4.6	219	15.4	65
Top 6 (+ CR-NET)	96.6%	5.3	189	20.7	48
Top 7 (+ STAR-Net)	96.9%	7.1	141	27.8	36
Top 8 (+ ViTSTR-Base)	96.9%	7.3	137	35.0	29
Top 9 (+ GRCNN)	97.1%	8.5	117	43.6	23
Top 10 (+ RARE)	97.1%	13.0	77	56.6	18
Top 11 (+ R <sup>2</sup> AM)	97.1%	15.9	63	72.5	14
Top 12 (+ TRBA)	97.1%	16.9	59	89.4	11

Remarkably, fusing the outputs of the three fastest models results in a lower recognition rate (91.1%) than using the best model alone (92.4%). Nevertheless, as more methods are included in the ensemble, the gap reduces considerably. From this observation, we can infer that if attaining the utmost recognition rate across various scenarios is not imperative, it becomes more advantageous to combine fewer but faster models, as long as they perform satisfactorily individually. According to Table 6, combining 4–6 fast models appears to be the optimal choice for striking a better balance between speed and accuracy.

## 5 Conclusions and Future Work

This paper examines the potential improvements in LPR results by fusing the outputs from multiple recognition models. Distinguishing itself from prior studies, our research explores a wide range of models and datasets in the experiments. We combined the outputs of different models through straightforward approaches such as selecting the most confident prediction or through majority vote (both at

sequence and character levels), demonstrating the substantial benefits of fusion approaches in both intra- and cross-dataset experimental setups.

In the traditional intra-dataset setup, where we explored eight datasets, the mean recognition rate experienced a significant boost, rising from 92.4% achieved by the best model individually to 97.6% when leveraging model fusion. Essentially, we demonstrate that fusing multiple models reduces considerably the likelihood of obtaining subpar performance on a particular dataset. In the more challenging cross-dataset setup, where we explored four datasets, the mean recognition rate increased from 87.6% to rates surpassing 90%. Notably, the optimal fusion approach in both setups was via majority vote at sequence level.

We also conducted an evaluation to analyze the speed/accuracy trade-off in the final approach by varying the number of models included in the ensemble. For this assessment, we ranked the models in two distinct ways: one based on their recognition results and the other based on their efficiency. The findings led us to the conclusion that for applications where the recognition task can tolerate some additional time, though not excessively, an effective strategy is to combine 4–6 fast models. Employing this approach significantly enhances recognition results while maintaining the system’s efficiency at an acceptable level.

In future work, we aim to delve into more sophisticated approaches for combining the recognition models. One avenue involves training a network that takes the outputs of multiple models as input and learns to generate the final prediction. Another viable approach involves considering the input data rather than directly combining the predictions, with the goal of selecting the optimal model for each specific data point. This can be accomplished using a gating network that determines which model should influence the output the most.

## ACKNOWLEDGMENTS

We thank the support of NVIDIA Corporation with the donation of the Quadro RTX 8000 GPU used for this research.

## References

1. Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: International Conference on Document Analysis and Recognition. pp. 319–334 (2021)
2. Baek, J., et al.: What is wrong with scene text recognition model comparisons? Dataset and model analysis. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4714–4722 (2019)
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. pp. 1–14 (2020)
4. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: Large scale system for text detection and recognition in images. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 71–79 (2018)
5. Dlagnekov, L.: UCSD/Calit2 car license plate, make and model database. [http://vision.ucsd.edu/belongie-grp/research/carRec/car\\_data.html](http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html) (2005)

6. Fan, X., Zhao, W.: Improving robustness of license plates automatic recognition in natural scenes. *IEEE Transactions on Intelligent Transportation Systems*. **23**(10), 18845–18854 (2022)
7. Gao, Y., et al.: GroupPlate: Toward multi-category license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*. **24**(5), 5586–5599 (2023)
8. Garcia-Bordils, S., Mafla, A., Biten, A.F., Nuriel, O., Mazor, S., Litman, R., Karatzas, D.: Out-of-vocabulary challenge report. In: *European Conference on Computer Vision (ECCV), TiE: Text in Everything Workshop*. pp. 1–17 (2022)
9. Gonçalves, G.R., Silva, S.P.G., Menotti, D., Schwartz, W.R.: Benchmark for license plate character segmentation. *Journal of Electronic Imaging*. **25**(5), 053034 (2016)
10. Gonçalves, G.R., et al.: Real-time automatic license plate recognition through deep multi-task networks. In: *Conference on Graphics, Patterns and Images (SIB-GRAPI)*. pp. 110–117 (Oct 2018)
11. Gonçalves, G.R., et al.: Multi-task learning for low-resolution license plate recognition. In: *Iberoamerican Congress on Pattern Recognition*. pp. 251–261 (Oct 2019)
12. Gong, Y., Deng, L., Tao, S., Lu, X., Wu, P., Xie, Z., Ma, Z., Xie, M.: Unified Chinese license plate detection and recognition with high efficiency. *Journal of Visual Communication and Image Representation*. **86**, 103541 (2022)
13. Henry, C., Ahn, S.Y., Lee, S.: Multinational license plate recognition using generalized character sequence detection. *IEEE Access*. **8**, 35185–35199 (2020)
14. Hsu, G.S., Ambikapathi, A., Chung, S.L., Su, C.P.: Robust license plate detection in the wild. In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. pp. 1–6 (Aug 2017)
15. Hsu, G.S., Chen, J.C., Chung, Y.Z.: Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*. **62**(2), 552–561 (2013)
16. Izidio, D.M.F., et al.: An embedded automatic license plate recognition system using deep learning. *Design Automation for Embedded Systems*. **24**(1), 23–43 (2020)
17. Kabiraj, A., Pal, D., Ganguly, D., Chatterjee, K., Roy, S.: Number plate recognition from enhanced super-resolution using generative adversarial network. *Multimedia Tools and Applications*. **82**(9), 13837–13853 (2023)
18. Ke, X., Zeng, G., Guo, W.: An ultra-fast automatic license plate recognition approach for unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*. **24**(5), 5172–5185 (2023)
19. Laroca, R., Araujo, A.B., Zanlorensi, L.A., De Almeida, E.C., Menotti, D.: Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach. *IEEE Access*. **9**, 67569–67584 (2021)
20. Laroca, R., Cardoso, E.V., Lucio, D.R., Estevam, V., Menotti, D.: On the cross-dataset generalization in license plate recognition. In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. pp. 166–178 (Feb 2022)
21. Laroca, R., et al.: A robust real-time automatic license plate recognition based on the YOLO detector. In: *International Joint Conference on Neural Networks (IJCNN)*. pp. 1–10 (July 2018)
22. Laroca, R., et al.: An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intelligent Transport Systems*. **15**(4), 483–503 (2021)
23. Lee, C., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2231–2239 (2016)
24. Lee, Y., et al.: License plate detection via information maximization. *IEEE Transactions on Intelligent Transportation Systems*. **23**(9), 14908–14921 (2022)

25. Liu, Q., Chen, S.L., Li, Z.J., Yang, C., Chen, F., Yin, X.C.: Fast recognition for multidirectional and multi-type license plates with 2D spatial attention. In: Intl. Conference on Document Analysis and Recognition (ICDAR). pp. 125–139 (2021)
26. Liu, W., Chen, C., Kwan-Yee K. Wong, Z.S., Han, J.: STAR-Net: A spatial attention residue network for scene text recognition. In: British Machine Vision Conference (BMVC). pp. 1–13 (Sept 2016)
27. Mokayed, H., Shivakumara, P., Woon, H.H., Kankanhalli, M., Lu, T., Pal, U.: A new DCT-PCM method for license plate number detection in drone images. *Pattern Recognition Letters*. **148**, 45–53 (2021)
28. Nascimento, V., et al.: Super-resolution of license plate images using attention modules and sub-pixel convolution layers. *Computers & Graphics*. **113**, 69–76 (2023)
29. OpenALPR: OpenALPR-EU dataset. <https://github.com/openalpr/benchmarks/tree/master/endtoend/eu> (2016)
30. Polikar, R.: Ensemble learning. *Ensemble Machine Learning: Methods and Applications*. Springer New York. pp. 1–34 (2012)
31. Schirmacher, F., Lorch, B., Maier, A., Riess, C.: Benchmarking probabilistic deep learning methods for license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*. pp. 1–14 (2023), In Press (Early Access).
32. Shi, B., Bai, X., Yao, C.: 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**(11), 2298–2304 (Nov 2017)
33. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4168–4176 (2016)
34. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: *European Conf. on Computer Vision (ECCV)*. pp. 593–609 (2018)
35. Silva, S.M., Jung, C.R.: Real-time license plate detection and recognition using deep convolutional neural networks. *Journal of Visual Communication and Image Representation*. p. 102773 (2020)
36. Silva, S.M., Jung, C.R.: A flexible approach for automatic license plate recognition in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*. **23**(6), 5693–5703 (2022)
37. Špaňhel, J., Sochor, J., Juránek, R., Herout, A.: Geometric alignment by deep learning for recognition of challenging license plates. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. pp. 3524–3529 (Nov 2018)
38. Špaňhel, J., et al.: Holistic recognition of low quality license plates by CNN using track annotated data. In: *IEEE International Conference on Advanced Video and Signal Based Surveillance*. pp. 1–6 (Aug 2017)
39. Srebrić, V.: EnglishLP database. [https://www.zemris.fer.hr/projects/LicensePlates/english/baza\\_slika.zip](https://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip) (2003)
40. Terven, J., Cordova-Esparza, D.: A comprehensive review of YOLO: From YOLOv1 and beyond. *arXiv preprint arXiv:2304.00501*. pp. 1–33 (2023)
41. Wang, J., Hu, X.: Gated recurrent convolution neural network for OCR. In: *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*. p. 334–343 (2017)
42. Wang, Y., Bian, Z.P., Zhou, Y., Chau, L.P.: Rethinking and designing a high-performing automatic license plate recognition approach. *IEEE Transactions on Intelligent Transportation Systems*. **23**(7), 8868–8880 (2022)
43. Weber, M.: Caltech Cars. <https://data.caltech.edu/records/20084> (1999)
44. Yuan, Y., Zou, W., Zhao, Y., Wang, X., Hu, X., Komodakis, N.: A robust and efficient approach to license plate detection. *IEEE Transactions on Image Processing*. **26**(3), 1102–1114 (2017)



45. Zeni, L.F., Jung, C.: Weakly supervised character detection for license plate recognition. In: Conference on Graphics, Patterns and Images. pp. 218–225 (2020)
46. Zhang, L., Wang, P., Li, H., Li, Z., Shen, C., Zhang, Y.: A robust attentional framework for license plate recognition in the wild. *IEEE Transactions on Intelligent Transportation Systems*. **22**(11), 6967–6976 (2021)
47. Zhang, M., Liu, W., Ma, H.: Joint license plate super-resolution and recognition in one multi-task GAN framework. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1443–1447 (April 2018)
48. Zhou, W., et al.: Principal visual word discovery for automatic license plate detection. *IEEE Transactions on Image Processing*. **21**(9), 4269–4279 (Sept 2012)
49. Zou, Y., Zhang, Y., Yan, J., Jiang, X., Huang, T., Fan, H., Cui, Z.: A robust license plate recognition model based on Bi-LSTM. *IEEE Access*. **8**, 211630–211641 (2020)