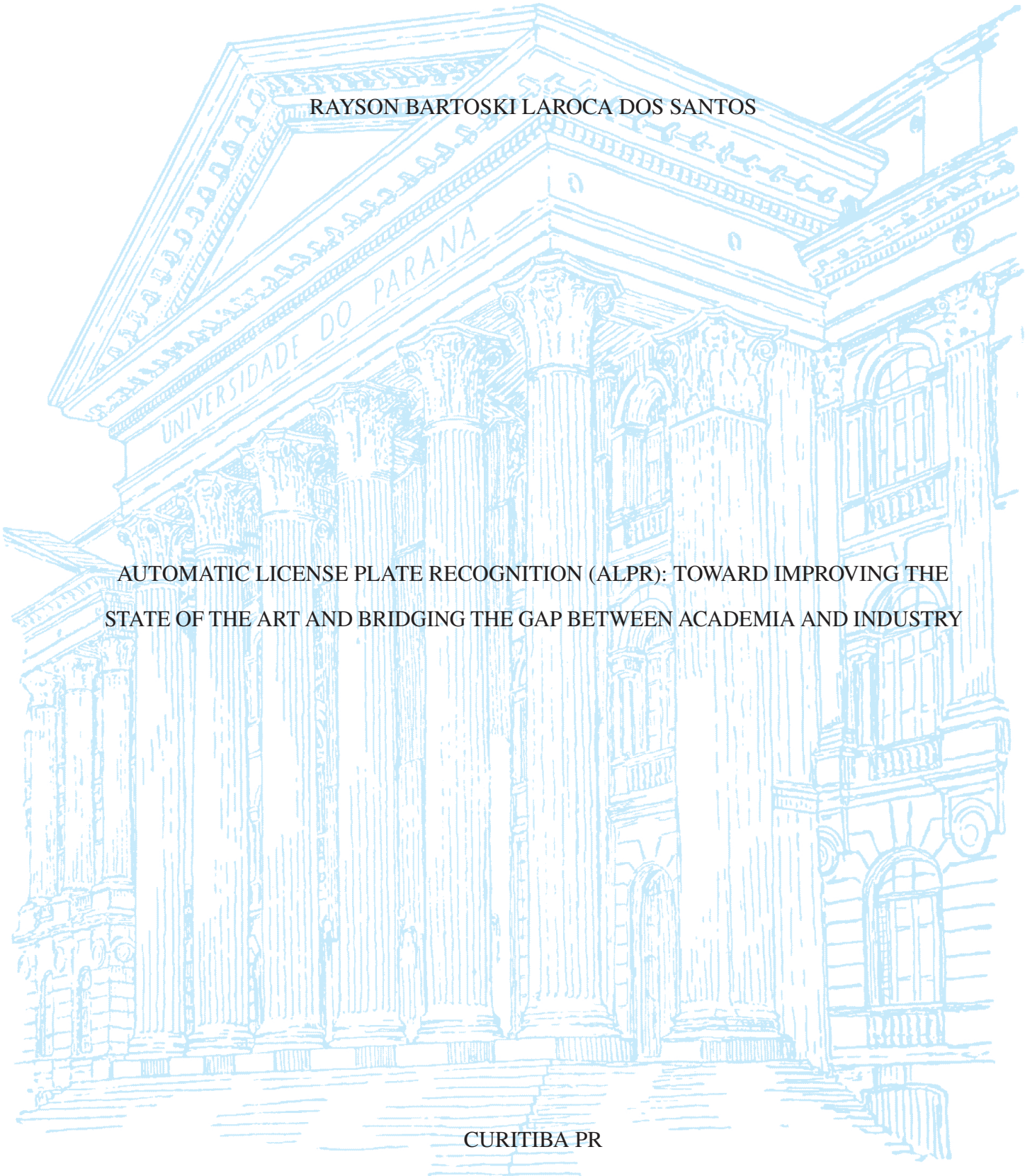
UNIVERSIDADE FEDERAL DO PARANÁ

RAYSON BARTOSKI LAROCA DOS SANTOS



AUTOMATIC LICENSE PLATE RECOGNITION (ALPR): TOWARD IMPROVING THE STATE OF THE ART AND BRIDGING THE GAP BETWEEN ACADEMIA AND INDUSTRY

CURITIBA PR

2024

RAYSON BARTOSKI LAROCA DOS SANTOS

AUTOMATIC LICENSE PLATE RECOGNITION (ALPR): TOWARD IMPROVING THE STATE OF THE ART AND BRIDGING THE GAP BETWEEN ACADEMIA AND INDUSTRY

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: David Menotti.

Coorientador: Rodrigo Minetto.

CURITIBA PR

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Santos, Rayson Bartoski Laroca dos
Automatic license plate recognition (ALPR) : toward improving the state of the art and bridging the gap between academia and industry / Rayson Bartoski Laroca dos Santos. – Curitiba, 2024.
1 recurso on-line: PDF.

Tese (doutorado) – Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática. Defesa: Curitiba, 25/04/2024.
Orientador: David Menotti
Coorientador: Rodrigo Minetto

1. Sistemas de reconhecimento de padrões. 2. Placas de automóveis. 3. Automóveis - Identificação I. Menotti, David. II. Minetto, Rodrigo. III. Título

CDD 006.4

Selma Regina Ramalho Conte
Bibliotecária
CRB-9/888

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **RAYSON BARTOSKI LAROCA DOS SANTOS** intitulada: **Automatic License Plate Recognition (ALPR): Toward Improving the State of the Art and Bridging the Gap Between Academia and Industry**, sob orientação do Prof. Dr. DAVID MENOTTI GOMES, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 25 de Abril de 2024.

Assinatura Eletrônica
26/04/2024 11:51:23.0
DAVID MENOTTI GOMES
Presidente da Banca Examinadora

Assinatura Eletrônica
26/04/2024 12:07:40.0
EDUARDO JOSÉ DA SILVA LUZ
Avaliador Externo (UNIVERSIDADE FEDERAL DE OURO PRETO)

Assinatura Eletrônica
27/04/2024 20:22:58.0
EDUARDO TODT
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
26/04/2024 13:41:54.0
CLAUDIO ROSITO JUNG
Avaliador Externo (UNIVER. FEDERAL DO RIO GRANDE DO SUL)

Assinatura Eletrônica
26/04/2024 12:49:30.0
RODRIGO MINETTO
Coorientador(a) (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)

To my parents and grandparents.

ACKNOWLEDGMENTS

I express my heartfelt gratitude to my advisor, Professor David Menotti, for his invaluable support and understanding, especially during the trying times of the COVID-19 pandemic. Professor David has shaped my perspective as a researcher in many fundamental ways. He provided me with the freedom to explore new ideas and experiment with innovative methods, all while offering critical insights that steered the direction of my research.

Professor Rodrigo Minetto, my co-advisor, deserves special thanks for his insightful guidance throughout my studies. I extend my gratitude to the esteemed members of my thesis committee, Professors Claudio Jung, Eduardo Luz, and Eduardo Todt, for their contributions to the refinement of this document. I also want to thank Professor Lucas Ferrari's involvement in my qualifying exam, which significantly contributed to the development of this work.

My appreciation goes to the co-authors of my published articles for their collaboration and expertise. I also want to thank the professors and colleagues from the Department of Informatics at UFPR for creating a collaborative environment that enriched my academic experience.

I am deeply grateful to my fiancée, Maria Karolina Ramos, with whom I have shared over a decade of companionship. Karol stood by me during challenging times, providing much-needed perspective and encouragement. I fully recognize the crucial role she played in enabling me to persevere until the completion of this journey. This work is also dedicated to her.

Last but not least, I want to thank my parents, grandparents and brothers. While my grandparents could not witness the completion of my PhD studies, they would undoubtedly have taken great pride in my accomplishments. My family has been a constant source of motivation throughout my academic journey, offering unwavering support. I am deeply thankful for their unconditional love, which has been instrumental in shaping the individual I am today.

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) (Social Demand Program). I gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro RTX 8000 GPU used for this research.

"Most good ideas [towards human-level AI] will come from academia, even if the most impressive applications come from industry."

— Yann LeCun

RESUMO

O reconhecimento automático de placas de veículos (ALPR) tem sido um tópico de pesquisa frequente devido às suas amplas aplicações práticas, incluindo cobrança automática de pedágios e aplicação das leis de trânsito. Apesar do progresso considerável no estado da arte nos últimos anos, várias questões persistem em aberto neste domínio. Esta tese investiga o potencial para avanços significativos no ALPR ao investigar e abordar meticulosamente essas questões, em vez de focar no aumento do número de imagens reais de treinamento, na proposta de descritores inovadores, ou na busca extensiva por melhores arquiteturas de modelos. Nossa pesquisa começa endereçando a falta de atenção dada às imagens contendo placas Mercosul, motocicletas, e placas com duas linhas de caracteres através da criação de um conjunto de dados dedicado (RodoSol-ALPR) e da condução de uma série de experimentos com ele. Nossos experimentos ressaltam a importância deste conjunto de dados para o reconhecimento robusto de placas Mercosul e de placas com duas linhas de caracteres, já que modelos de reconhecimento óptico de caracteres (OCR) treinados em outros conjuntos de dados não conseguem ultrapassar uma taxa de reconhecimento de 70% em seu conjunto de teste. Posteriormente, apresentamos melhorias substanciais no desempenho do ALPR de ponta a ponta ao mesclar a saída de vários modelos de OCR e combinar várias metodologias de geração de dados sintéticos. Notavelmente, a utilização extensiva de dados sintéticos leva a resultados estado-da-arte em diversos conjuntos de dados e desempenha um papel fundamental na superação de desafios causados pela disponibilidade limitada de dados de treinamento. Esta tese também identifica questões críticas na avaliação de sistemas para o ALPR. Revelamos que os protocolos de avaliação estabelecidos não levam em conta as quase duplicatas nos conjuntos de treinamento e teste, dificultando o desenvolvimento e a aceitação de modelos mais eficientes que tenham fortes habilidades de generalização mas não memorizam duplicatas tão bem quanto outros modelos. Por fim, contextualizamos o problema do viés de conjunto de dados no domínio do ALPR, aumentando a conscientização sobre suas possíveis consequências. A identificação destas questões enfatiza a importância da realização de experimentos *cross-dataset*, uma vez que estes fornecem uma melhor indicação de generalização do que experimentos *intra-dataset*. Uma maior adoção de avaliações *cross-dataset* tem o potencial de reduzir a lacuna entre os resultados relatados no meio acadêmico e os alcançados na indústria.

Palavras-chave: Reconhecimento Automático de Placas de Veículos, Generalização *Cross-Dataset*, Viés de Conjunto de Dados, Layout Mercosul, Fusão de Modelos, Quase Duplicatas, Conjuntos de Dados Públicos, Dados Sintéticos.

ABSTRACT

Automatic License Plate Recognition (ALPR) has been a frequent research topic due to its wide-ranging practical applications, including automatic toll collection and traffic law enforcement. Despite the considerable progress in the state of the art driven by deep learning and the increasing availability of public datasets, several open issues persist within the ALPR domain. This thesis investigates the potential for significant advancements in ALPR by meticulously identifying and addressing these issues, rather than focusing on increasing the number of real training images, designing groundbreaking descriptors, or extensively searching for better model architectures. Our research begins by tackling the lack of attention given to images featuring Mercosur License Plates (LPs), motorcycles, and two-row LPs by creating a dedicated dataset (RodoSol-ALPR) and conducting a series of experiments using it. Our experiments underscore the importance of the RodoSol-ALPR dataset for robust recognition of Mercosur and two-row LPs, as Optical Character Recognition (OCR) models trained on alternative datasets fail to surpass a 70% recognition rate on its test set. Subsequently, we showcase substantial improvements in end-to-end ALPR performance by fusing the outputs of multiple OCR models and combining various synthetic data generation methodologies. Notably, the extensive use of synthetic data leads to state-of-the-art results across diverse datasets and plays a pivotal role in overcoming challenges caused by limited training data availability. This thesis also identifies critical issues in the assessment of ALPR systems. We reveal that established evaluation protocols have failed to account for near-duplicates within training and test sets, hindering the development and acceptance of more efficient models that have strong generalization abilities but do not memorize duplicates as well as other models. Finally, we contextualize the dataset bias problem within the License Plate Recognition (LPR) domain, raising awareness about its potential consequences and discussing the subtle ways this bias may have crept into existing datasets. Identifying these issues emphasizes the importance of conducting cross-dataset experiments, as they provide a better indication of generalization than intra-dataset ones. This shift toward cross-dataset setups has the potential to bridge the gap between results reported in academia and those achieved in industry.

Keywords: Automatic License Plate Recognition, Cross-Dataset Generalization, Dataset Bias, Mercosur Layout, Model Fusion, Near-Duplicates, Public Datasets, Synthetic Data.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AC	Access Control
AI	Artificial Intelligence
ALPR	Automatic License Plate Recognition
AMR	Automatic Meter Reading
AP	Average Precision
API	Application Programming Interface
AUC	Area Under the Curve
Bi-LSTM	Bi-directional Long Short-Term Memory
BRNN	Bidirectional Recurrent Neural Network
CCA	Connected Component Analysis
cGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CRNN	Convolutional Recurrent Neural Network
CTC	Connectionist Temporal Classification
DCGAN	Deep Convolutional Generative Adversarial Network
DCT	Discrete Cosine Transform
DENATRAN	National Traffic Department of Brazil
FID	Fréchet Inception Distance
FN	False Negative
FP	False Positive
FPS	Frames Per Second
GAN	Generative Adversarial Network
GAT	Graph Attention Network
GPU	Graphics Processing Unit
GRCNN	Gated Recurrent Convolution Neural Network
GRU	Gated Recurrent Unit
GT	Ground Truth
HC	Highest Confidence
IIA	Iterated Integrated Attributions
IoU	Intersection over Union
IS	Inception Score
IWPOD-NET	Improved Warped Planar Object Detection Network
LE	traffic Law Enforcement
LODO	Leave-One-Dataset-Out
LP	License Plate
LPD	License Plate Detection
LPR	License Plate Recognition
LSTM	Long Short-Term Memory
mAP	mean Average Precision
MLP	Multilayer Perceptron
MSE	Mean Squared Error

MV	Majority Vote
MVCP	Majority Vote by Character Position
NLP	Natural Language Processing
NME	Normalization Mean Error
NMS	Non-Maximum Suppression
OCR	Optical Character Recognition
PReLU	Parametric ReLU
PSENet	Progressive Scale Expansion Network
R^2AM	Recursive Recurrent neural networks with Attention Modeling
RARE	Robust text recognizer with Automatic REctification
RCNN	Recurrent Convolutional Neural Network
ReLU	Rectified Linear Unit
ResNet	Residual Network
ROC	Receiver Operating Characteristic
RodoSol	*Rodovia do Sol*
RP	Road Patrol
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SPP	Spatial Pyramid Pooling
STAR-Net	SpaTial Attention Residue Network
STN	Spatial Transformer Network
SVM	Support Vector Machine
TIR	Thermal Infrared
TP	True Positive
TPS	Thin-Plate Splines
TRBA	TPS-ResNet-BiLSTM-Attention
VAE	Variational Autoencoder
ViT	Vision Transformer
VOC	Visual Object Classes
WPOD-NET	Warped Planar Object Detection Network
WSCD	Weakly Supervised Character Detection

CONTENTS

1. INTRODUCTION

The global automotive industry's sales volume has recently rebounded to pre-pandemic levels (Statista, 2024; ING Economics, 2024). In addition to bringing convenience to owners, vehicles also significantly modify the urban environment, posing pollution, privacy and security challenges, especially in large urban centers. The continuous monitoring of vehicles through computational techniques is of paramount importance and has consequently become a prevalent area of research. In this context, Automatic License Plate Recognition (ALPR) systems stand out.

ALPR systems leverage image processing and pattern recognition techniques to detect and recognize License Plates (LPs) from images or videos. Some practical applications for an ALPR system are road traffic monitoring, toll collection, and vehicle access control in restricted areas (Anagnostopoulos et al., 2008; Du et al., 2013; Weihong and Jiaoyang, 2020).

In the deep learning era, ALPR systems typically include two stages: License Plate Detection (LPD) and License Plate Recognition (LPR). As depicted in Figure 1.1, the former stage involves locating the LP regions within the input image, while the latter refers to identifying the characters on those LPs. Both of these stages are crucial to the overall system performance and must be executed close to perfection, as (i) a failure in LPD often leads to subsequent failures in LPR, and (ii) a single incorrectly recognized character can result in the incorrect identification of the vehicle (Gonçalves et al., 2016b; Shashirangana et al., 2022; Ding et al., 2024).

Figure 1.1: A typical ALPR system. It is divided into two stages: LPD and LPR. The former stage refers to locating the LPs within the input image, while the latter refers to identifying the characters on those LPs.

ALPR systems have exhibited remarkable performance on LPs from multiple regions due to advances in deep learning and the increasing availability of annotated datasets (Henry et al., 2020; Silva and Jung, 2022; Liu et al., 2024b). Despite the considerable progress in the state of the art, many issues remain unresolved within the ALPR domain.

1.1 Problem Statement

This section outlines the key problems identified in the literature, which motivate our research.

Evaluation Protocols

In the past, the evaluation of ALPR systems used to be done within individual datasets. This involved training and testing the proposed methods on different subsets from the same dataset, with the models being trained and tested independently for each dataset. However, a recent shift has occurred due to the time-consuming nature of training deep learning models, especially on low- and mid-end Graphics Processing Units (GPUs). Researchers have embraced a new protocol where the models are trained once on the union of the training images from the selected datasets and then evaluated separately on the respective test sets (Laroca et al., 2021b; Qin and Liu, 2022; Pattanaik and Balabantaray, 2023). This protocol is hereinafter referred to

as *traditional split*. Despite using disjoint subsets for training and testing, such a protocol does not indicate whether the evaluated models have good generalization ability (i.e., whether they perform well on images from different scenarios), mainly due to domain divergence and data selection bias (Torralba and Efros, 2011; Zhang et al., 2019a; Fabbrizzi et al., 2022).

In this regard, many computer vision researchers have carried out cross-dataset experiments – where training and testing data come from different sources – to assess whether the proposed models perform well on data from an unknown domain (Ashraf et al., 2018; Ma et al., 2021; Estevam et al., 2024). Nevertheless, to our knowledge, research in ALPR lacks in-depth exploration of such experimental settings. This contrasts with the fact that real-world deployments often involve installing new cameras without retraining existing models. Adopting a *leave-one-dataset-out* evaluation protocol would effectively simulate this specific scenario and provide a more robust assessment of the models' generalizability.

Our practical experience has revealed that even when training the models on images from the same scenario (as in the traditional-split protocol), the accuracy levels observed in real-world deployments often fall short of those reported in academic studies. One possible explanation for this discrepancy is *dataset bias*, a well-recognized issue in the computer vision community (Ashraf et al., 2018; Jaipuria et al., 2022; Hort et al., 2023). Essentially, models inadvertently learn idiosyncrasies unique to each dataset alongside fundamental task-related knowledge. We also have discovered that the protocols traditionally adopted for splitting the images in public datasets into training and test sets do not account for the same vehicle or LP appearing in multiple images. Hence, distinct yet highly similar images of the same vehicle or LP may exist in both the training and test sets. Somewhat alarmingly, these issues (dataset bias and *near-duplicates* within the training and test sets) have gone unnoticed in the ALPR literature.

Diverse LP Layouts

Increased mobility and internationalization set new challenges for developing effective traffic monitoring and control systems. This is particularly true for ALPR systems, which must handle LPs from multiple regions with different character sets and syntax (Mecocci and Tommaso, 2006; Anagnostopoulos et al., 2008; Lubna et al., 2021). As shown in Figure 1.2, even LPs from the same country can vary considerably. For example, in the United States, many states allow *specialty* LPs showcasing the emblems of colleges, universities, professional sports teams, or other organizations. Individuals can also customize the arrangement of letters and digits for an extra fee (*vanity* LPs) (Guggenheim and Silversmith, 2000). Despite this variety, most ALPR systems presented in the literature were tailored to handle a single LP style (e.g., single-row blue LPs from mainland China). This limitation has been increasingly pointed out in recent research (Zeni and Jung, 2020; Silva and Jung, 2022; Gao et al., 2023). Although some authors claimed that their approaches could be extended with minor modifications to detect and recognize LPs from another region (Liu and Chang, 2019; Wang et al., 2022a; Rao et al., 2024), adapting layout-specific approaches to handle multiple LP layouts – with a similar degree of robustness – can be quite challenging or even unfeasible (Gao et al., 2020b; Laroca et al., 2021b).

Mercosur LPs

Mercosur, short for *Mercado Común del Sur* (Southern Common Market in Spanish), is an economic and political bloc comprising Argentina, Brazil, Paraguay and Uruguay[1]. These countries have collectively adopted a standardized format for LPs on newly purchased vehicles, as shown in Figure 1.3, drawing inspiration from the integrated system long adopted by member countries of the European Union. Despite the adoption of this new layout across all countries in the bloc, there is still no public dataset for ALPR with images of Mercosur LPs.

[1] Venezuela is currently suspended, and Bolivia is in the process of accession (MERCOSUR, 2024).

Figure 1.2: Examples of different LP styles in the United States. One can infer that it would be impractical to train an ALPR system specifically for each LP style. Image reproduced from `http://www.ashtonrose.org/blog/new-north-dakota-license-plate` (available via `http://web.archive.org/`).

Figure 1.3: The new standard of LPs adopted by Mercosur countries. This standard allows for any combination of letters and digits on the LP. The initial pattern adopted by each member country is shown above.

Motorcycles and Two-Row LPs

Motorcycles constitute a major form of transportation in urban areas, especially in developing nations (Hsu et al., 2015; Oliveira et al., 2021; Yuniaristanto et al., 2024). For instance, motorcycles make up over 90% of traffic in Vietnam (Nguyen-Phuoc et al., 2024) and 28% of all vehicles in Brazil (Senatran, 2024). This makes it crucial for ALPR systems to handle motorcycle images very well. Startlingly, motorcycles have been largely overlooked in ALPR research. While most researchers have used datasets without motorcycle images to evaluate their methods (Weihong and Jiaoyang, 2020; Lubna et al., 2021), there are several works where all images of motorcycles were explicitly excluded from the experiments (Gonçalves et al., 2018; Yonetsu et al., 2019; Fernandes et al., 2020). The lack of attention toward motorcycles in the ALPR literature is mainly because LPs of motorcycles usually have two rows of characters, which create difficulties for sequential/recurrent-based methods (Zeni and Jung, 2020; Xu et al., 2022; Chen et al., 2023), and also because they are generally smaller in size (with smaller and closely spaced characters) and are often tilted, further complicating recognition efforts.

Public Datasets

In this sense, there is a great demand for a publicly available dataset for end-to-end ALPR that contains the same number of images of cars and motorcycles, ensuring that both vehicle types receive equal importance during experimental evaluations. Ideally, the dataset should also encompass an equal distribution of LPs with one and two rows of characters. As highlighted by Ponce et al. (2006), the results may be biased when there are many more images for some "easy" samples (e.g., cars with single-row LPs) than for some "hard" ones (e.g., motorcycles with two-row LPs). For simplicity and in line with common practice in the literature, in this work "car" refers to any vehicle with four wheels or more (e.g., passenger cars, vans, buses, trucks, among others), whereas "motorcycle" refers to both motorcycles and motorized tricycles.

Synthetic LP Images

In the regime where labeled data is expensive (Björklund et al., 2019; Han et al., 2020; Gao et al., 2023) and privacy concerns are growing (Chan et al., 2020; Kong et al., 2021; Trinh et al., 2023), researchers would also benefit significantly from an approach capable of generating fully labeled images of LPs from diverse regions and styles. While recent studies have delved into the creation of synthetic LP images to enhance LPR performance, there are several limitations within these efforts, as elaborated in the following paragraph.

In addition to most works focusing on LPs from a single region, as discussed earlier, existing studies have predominantly employed a single methodology to generate synthetic LPs. This leaves open questions regarding the potential for significantly enhanced outcomes through the integration of data generated from various methodologies. Moreover, current research has mostly explored unpaired image-to-image translation methods (e.g., CycleGAN) using a large number of real images for training (100k+), without addressing how to achieve similar results with a limited number of real images for training. This need for many images limits the application of these methods since there are not always a large number of images available for each LP layout (Han et al., 2020; Laroca et al., 2021b; Yang et al., 2023). Finally, the assessment of synthetic data generation methods has mainly relied on the performance of individual Optical Character Recognition (OCR) models, overlooking the possibility that images created using a particular method may disproportionately favor certain models over others.

OCR Model Fusion

Regarding OCR models, previous research has shown that different models perform with varying degrees of robustness on different datasets (Zeni and Jung, 2020; Mokayed et al., 2021; Al-batat et al., 2022). Each dataset poses distinct challenges, such as diverse LP layouts and varying tilt ranges. As a result, a model that performs exceptionally well on one dataset may produce subpar results on another. This highlights the potential for significantly enhancing LPR results by fusing the outputs of diverse OCR models. The extent of this improvement and the optimal number and selection of models required remain unaddressed in the current literature.

Summary

The evaluation protocols traditionally adopted to assess ALPR systems fail to accurately indicate these systems' out-of-domain robustness. Moreover, they allow the same vehicle or LP to appear in both the training and test sets, potentially leading to skewed outcomes, even in intra-dataset evaluations. Current research has primarily focused on designing ALPR systems tailored to a single LP layout, neglecting the challenges of increased mobility and internationalization. There is a clear demand for a publicly available dataset that incorporates Mercosur LPs and includes an equal distribution of vehicle types (cars and motorcycles) and LP configurations (one- and two-row LPs). The ability to synthesize diverse and high-quality LP images is highly desirable to reduce the reliance on private datasets and address growing privacy concerns. Current methods for synthetic LP generation have several limitations, including a narrow focus on LP styles from specific regions, a lack of exploration of combining data generation methodologies, and the requirement for many real training images. Finally, the potential for improved performance by combining the output of multiple OCR models remains largely unexplored.

1.2 Hypothesis and Research Questions

The main hypothesis of this research is:

Hypothesis

It is possible to significantly improve the state of the art in Automatic License Plate Recognition (ALPR) *without increasing the number of real training images, designing groundbreaking descriptors, or extensively searching for better model architectures.*

More specifically, we firmly believe we can considerably improve the state of the art in ALPR by focusing on aspects often overlooked in the literature. These aspects include but are not limited to (i) addressing the lack of attention given to images featuring Mercosur LPs, motorcycles, and two-row LPs through the creation of a dedicated dataset, (ii) leveraging

fusion approaches to enhance LPR performance across various scenarios, and (iii) developing a Generative Adversarial Network (GAN)-based methodology for synthesizing fully-labeled images of LPs from diverse regions and styles. Integrating this methodology with others can lead to improved LPR performance and reduce the reliance on large volumes of real training data.

We are also certain that moving beyond overly simplistic experimental setups will enable us to reveal limitations in current approaches and biases within established evaluation protocols. By actively addressing these issues, our proposed methods can yield results that not only surpass state-of-the-art approaches but also align more closely with those achieved in industry.

The following questions guide our research:

- What are the best practices for gathering images in real-world settings to build a dataset featuring images of vehicles with Mercosur LPs? What legal and ethical factors should be considered when collecting and selecting these images, such as the potential presence of identifiable faces within the images? What specific characteristics should this dataset possess, such as a balanced representation of cars and motorcycles, as well as an equal representation of Brazilian and Mercosur LPs[2]? What annotations must be provided for each image to enable the evaluation of ALPR systems in an end-to-end manner?

- Do current methods for detecting and recognizing LPs generalize well to unseen data? Why is it crucial to evaluate deep learning models on a range of datasets with varying characteristics? Is there an OCR model that stands out as superior across all datasets, regardless of their characteristics and the volume of training data? What influence does the proposed dataset have on the accurate recognition of Mercosur and two-row LPs?

- Can we significantly improve LPR results by combining the outputs of various OCR models? If so, to what extent can such enhancement be attained? Additionally, how many models and which specific ones should we explore for optimal results? When selecting models for the ensemble, should we prioritize their accuracy levels to maximize recognition performance, or would it be more advantageous to focus on faster models to strike a better balance between accuracy and speed in the final methodology?

- To what extent does combining real data with synthetic data generated through advanced techniques improve LPR accuracy compared to solely augmenting real data with standard transformations such as random perspective shifts, noise addition, and adjustments to brightness and contrast? What are the prevalent methodologies for generating synthetic LP images, and how do they stack up in terms of increasing LPR accuracy? Is there a synergistic effect from combining them, or is relying on a single method sufficient?

- Can a single generative model, trained with only a few hundred real images for each LP layout, produce fully-labeled images of LPs from diverse regions and styles? Can alternative methods for generating synthetic data be leveraged to overcome the scarcity of labeled paired data required to train an image-to-image translation model? How can character distortion or blending be mitigated during the generation of the LP images?

- To what degree can synthetic images, created using various methodologies, reduce the number of real images needed for effectively training OCR models? How do OCR models with similar performance fare when trained with reduced portions of the training

[2] To maintain consistency with previous works (Izidio et al., 2020; Oliveira et al., 2021; Silva and Jung, 2022), we refer to “Brazilian” as the layout used in Brazil before the adoption of the Mercosur layout.

set but supplemented with synthetic data? Do they exhibit comparable performance trends, or does one model outperform the other in such scenarios?

- Can we attain state-of-the-art performance without relying on heuristic rules or post-processing techniques to adjust the predictions based on expected patterns in particular LP layouts? How important is it to rectify (unwarp) the LPs before recognition for achieving these results? Which OCR model offers the optimal balance between speed and accuracy in each of the intra-dataset and cross-dataset protocols? Do models that strike a good speed/accuracy trade-off under the intra-dataset protocol maintain such equilibrium when applied to independent datasets? What specific characteristics enable these models (or hinder them) to sustain such balance?

- Do well-established partitions of ALPR datasets contain *near-duplicates* within their training and test sets? If so, how prevalent are these occurrences? How can these partitions be reworked to create *fair splits* that exclude duplicates while maintaining their key characteristics? Would LPR models trained and tested on these fair splits exhibit significantly higher error rates compared to those trained and tested on conventional partitions that include duplicates? What are the implications of such duplicates on the assessment and development of deep learning-based models for LPR?

- Are there identifiable signatures (bias) in public datasets that LPR models can exploit to identify from which dataset each LP image originates? If such biases are found, what impact have they had on the learning and evaluation of LPR models? Which strategies can be employed to mitigate dataset bias in upcoming data collections?

1.3 Objectives

This research aims to propel the field of *Automatic License Plate Recognition* forward. We seek to achieve this by identifying and meticulously analyzing the key limitations within the literature. By addressing these shortcomings, we aim to improve the state of the art and bridge the gap between the results reached in academia and industry. The specific objectives are as follows:

- To introduce a public dataset comprising many images of vehicles with Mercosur LPs acquired in real-world scenarios. We intend to meticulously curate this dataset to ensure a balanced distribution between images of cars and motorcycles, as well as Brazilian and Mercosur LPs. This approach aims to mitigate potential biases during the assessment of ALPR systems. Additionally, we plan to provide detailed annotations for each image, enabling a comprehensive end-to-end evaluation of ALPR systems;

- To draw researchers' attention to cross-dataset experiments since they better simulate real-world ALPR applications, where new cameras are regularly being installed in new locations without existing systems being retrained every time;

- To underscore the significant variations in how models perform on different datasets. We aim to emphasize the importance of evaluating models using a diverse range of datasets rather than relying on just a few that may not be fully representative;

- To explore potential improvements in LPR results by fusing the outputs from multiple OCR models. Our objective is to determine the most effective method of combining the chosen models, quantify the attainable performance gains, and find the optimal selection

of models for the ensemble, considering either the achieved recognition accuracy or the balance between speed and accuracy in the resulting methodology;

- To thoroughly evaluate synthetic data generation methodologies based on the average results achieved by diverse OCR models across various datasets. We aim to demonstrate the contributions of each synthesis method and assess how their combination could further enhance model performance compared to using a single methodology;
- To propose an end-to-end ALPR system that achieves state-of-the-art results on public benchmarks. The system must effectively handle challenges often found in real-world applications, such as diverse LP layouts, images with varying resolutions, LPs with different numbers of characters arranged in one or two rows, and scenarios where the LP characters are partially occluded. Ideally, the proposed system should demonstrate robustness to images captured in domains beyond those represented in the training set without requiring hundreds of thousands of real, human-labeled images for training;
- To highlight the fact that the evaluation protocols traditionally adopted to assess ALPR systems have historically failed to account for the possibility of the same vehicle or LP appearing in multiple images. We aim to understand how these near-duplicates have affected the performance evaluation of OCR models applied to LPR;
- To examine the issue of dataset bias in the LPR context, specifically investigating whether public datasets from the two dominant regions in the field have unique and identifiable signatures. Our goal is to bring attention to the significant ramifications of dataset bias in ALPR research, analyzing how this bias could have infiltrated these datasets and suggesting measures to identify and mitigate it in future data collection efforts;

1.4 Contributions

The contributions of this work can be summarized as follows:

- [**Chapter 4**] The first public dataset containing images of vehicles with Mercosur LPs. This dataset, named RodoSol-ALPR, is instrumental in enabling researchers to adapt and develop ALPR systems specifically for this new LP layout. RodoSol-ALPR facilitates fair comparisons between methods proposed in various studies due to its balanced distribution of images featuring cars and motorcycles, as well as one- and two-row LPs. Remarkably, access to the dataset has already been granted to 145 researchers from 42 countries around the world, as shown here. The dataset has already been explored in several works, including (Nascimento et al., 2023; Chen et al., 2023; Liu et al., 2024b);
- [**Chapter 5**] A comprehensive evaluation that highlights the importance of increasing the out-of-domain robustness of ALPR systems, particularly regarding LPR. We consider the proposed *traditional-split vs. leave-one-dataset-out* experimental setup to be a valid testbed for assessing the cross-dataset generalizability of forthcoming methods;
- [**Chapter 6**] A demonstration of the substantial benefits of fusion approaches to LPR performance, both in intra- and cross-dataset experimental setups. More specifically, we show that fusing multiple OCR models reduces considerably the likelihood of obtaining subpar performance on a particular scenario. This analysis includes a comparative assessment of distinct fusion methods and considers the speed/accuracy trade-off in the final approach by varying the number of models incorporated into the ensemble;

- [**Chapter 7**] A GAN-driven methodology for synthesizing fully-labeled images of LPs from diverse regions and styles. Despite being trained with only a few hundred real images per LP layout, it yields high-quality results. We are releasing a dataset with 300k LP images generated through this technique, which researchers can use for training and testing their OCR models. Such a dataset is of paramount importance as growing privacy concerns have inhibited the creation and availability of LP datasets in several regions;

- [**Chapter 7**] A thorough study on the effectiveness of multiple synthetic data generation methodologies, from creating template-based LP images using OpenCV to producing more realistic images using GANs, on the average performance across various OCR models. Our analysis goes beyond measuring the individual effectiveness of each methodology. We highlight the synergistic effect of combining them, leading to enhanced overall LPR performance. Furthermore, we demonstrate that synthetic data plays a crucial role in overcoming the challenges posed by limited training data availability;

- [**Chapter 7**] An end-to-end ALPR system that outperforms state-of-the-art approaches and established commercial solutions, excelling in both intra- and cross-dataset scenarios, despite being trained on a significantly smaller set of real images;

- [**Chapters 5 to 7**] Empirical evidence indicating that general-purpose detectors (e.g., YOLOv4 and its variants) can be reliably employed for LPD, even when dealing with images from unseen datasets. However, our experiments emphasize the importance of rectifying the LPs before feeding them into OCR models for optimal LPR performance;

- **[Chapters 5 to 7**] Several experimental findings that underscore the importance of comparing models across multiple datasets that have a wide variety in the way they were collected and that comprise images of various vehicle types and LP layouts;

- [**Chapter 8**] We reveal the large fraction of near-duplicates within the training and test sets of datasets widely adopted in ALPR research. Our findings suggest that such duplicates have biased the evaluation of deep learning-based models for LPR, potentially hindering the development and acceptance of more efficient models that have strong generalization abilities but do not memorize duplicates as well as other models. To address this issue, we have created and released *fair splits* for the two most popular datasets in the field. These new splits eliminate duplicates from the training and test sets while preserving the key characteristics of the original partitions as much as possible;

- [**Chapter 9**] A contextualization of the dataset bias problem within LPR, showing that a lightweight Convolutional Neural Network (CNN) can determine the source dataset of an LP image with over 95% accuracy. This level of accuracy far exceeds what would be expected by chance or human ability. In addition to raising awareness about the potential consequences of this bias, we discuss the subtle ways through which it may have crept into the datasets, paving the way for future research directions;

The works published during the PhD's studies are listed below. Publications directly stemming from this thesis are marked with a star (★). Works co-authored and those covering related fields are included if they have substantially contributed to the development of this research. As an example, insights into multi-task learning and the generation of synthetic data via character permutation were derived from (Gonçalves et al., 2019) (item 10). Another pertinent example is (Laroca et al., 2021a) (item 5), where we investigated image-based Automatic

Meter Reading (AMR) rather than ALPR. Two models that play a significant role in Chapter 7, CDCC-NET and Fast-OCR, were proposed in that work. CDCC-NET also inspires the creation of the DC-NET model in Chapter 9, while Fast-OCR is also explored in Chapters 5 and 6.

1. ★ 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)*, pp. 60-75, Nov 2023;

2. ★ 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)*, pp. 1-8, June 2023;

3. ★ 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)*, pp. 234-239, Oct 2022;

4. ★ 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)*, pp. 166-178, Feb 2022;

5. 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*, vol. 9, pp. 67569-67584, 2021;

6. 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*, vol. 15, no. 4, pp. 483-503, 2021;

7. 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*, vol. 113, pp. 69-76, 2023;

8. V. Nascimento, R. Laroca, J. A. Lambert, W. R. Schwartz, and D. Menotti, "Combining Attention Module and Pixel Shuffle for License Plate Super-Resolution" in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 228-233, Oct 2022;

9. I. O. de Oliveira, R. Laroca, D. Menotti, K. V. O. Fonseca, and R. Minetto, "Vehicle-Rear: A New Dataset to Explore Feature Fusion for Vehicle Identification Using Convolutional Neural Networks," *IEEE Access*, vol. 9, pp. 101065-101077, 2021;

10. 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)*, pp. 251-261, Oct 2019.

We are currently preparing two additional articles for submission to prestigious journals. The first article focuses on the fusion of real and synthetic data to enhance LPR, as discussed in Chapter 7. The second article is a comprehensive survey of public datasets for ALPR. Despite being near completion and containing numerous insights from this work, the second article has been omitted due to constraints within this document's scope.

1.5 Thesis Outline

The remainder of this thesis is structured as follows. Chapter 2 presents the theoretical foundation for the concepts used throughout this work. Chapter 3 reviews relevant research in the field. Chapter 4 introduces the RodoSol-ALPR dataset, the first to include Mercosur LPs. Chapter 5 covers our study on the cross-dataset generalization in LPR. Chapter 6 examines the potential for improving LPR results by combining the outputs from multiple OCR models. Chapter 7 delves into the integration of real and synthetic data to enhance LPR performance. Chapter 8 investigates the existence of near-duplicates within the training and test sets of datasets widely adopted in ALPR research. Chapter 9 situates the dataset bias problem in the LPR context. Finally, Chapter 10 lays out the conclusions of this work and proposes avenues for future research.

Please be aware that this thesis presents the research in a logical order that may differ from the original chronology. We have revised and reorganized several sections for improved coherence, and while we have carefully reviewed the manuscript, there may be minor inconsistencies.

2. THEORETICAL FOUNDATION

This chapter provides a concise theoretical foundation for the concepts explored in this work. We begin by describing the metrics commonly used to assess ALPR systems. As the LPD task comes down to detecting a single class of objects (LPs), many of these metrics were originally proposed for evaluating general object detectors. We then delve into the realm of deep learning, focusing specifically on CNNs and GANs. Finally, we discuss the concept of data augmentation.

2.1 Evaluation Metrics

The *precision* and *recall* evaluation metrics are commonly used in object detection (Everingham et al., 2010; Lin et al., 2014b; Padilla et al., 2020) and ALPR (Lu et al., 2021; Lee et al., 2022; Ding et al., 2024). These metrics are defined by comparing the areas covered by the ground truth and predicted bounding boxes, considering True Positives (TPs), False Positives (FPs), and False Negatives (FNs). Precision and recall can be formally expressed as follows:

$$precision = \frac{TP}{TP + FP}, \tag{2.1}$$

$$recall = \frac{TP}{TP + FN}. \tag{2.2}$$

In simpler terms, precision and recall are metrics that range from 0 to 1, with higher values indicating better performance. Precision measures the proportion of true positive results among all predictions, meaning a higher precision indicates fewer false positives. Conversely, recall measures the proportion of true positives that were correctly identified, meaning a higher recall indicates fewer false negatives. However, neither precision nor recall alone can accurately assess the match quality. For instance, recall can be artificially inflated by predicting numerous objects, even if many are incorrect (think of a system detecting many LPs in an image, even if most are not actually there). Conversely, a high precision rate can be achieved by being very selective, but at the cost of missing many correct identifications (imagine a system that only detects LPs with extremely high confidence, potentially missing many genuine ones).

The *F-measure* metric is defined as a harmonic mean of precision and recall. As shown in Equation 2.3, the most general form allows the differential weighting of precision and recall; however, they are commonly given equal weight (i.e., $\beta = 1$) (Powers, 2015). The *Average Precision* (AP) (Everingham et al., 2010) metric summarizes the shape of the precision/recall curve since it is defined as the average precision at a set of eleven equally spaced recall levels $[0, 0.1, \ldots, 1]$ (see Equation 2.4). Finally, the *mean Average Precision* (mAP) is calculated by taking the mean AP over all classes.

$$F\text{-}measure = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}, \tag{2.3}$$

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \max_{\tilde{r}:\tilde{r}>r} Precision(\tilde{r}). \tag{2.4}$$

A metric often used to assess the quality of predictions in object detection tasks is the *Intersection over Union* (IoU), also known as Jaccard index and Jaccard similarity coefficient, which can be expressed by the formula

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}, \tag{2.5}$$

where B_p and B_{gt} are the predicted and ground truth bounding boxes, respectively. Figure 2.1 illustrates this definition. The closer the IoU is to 1, the better the detection.

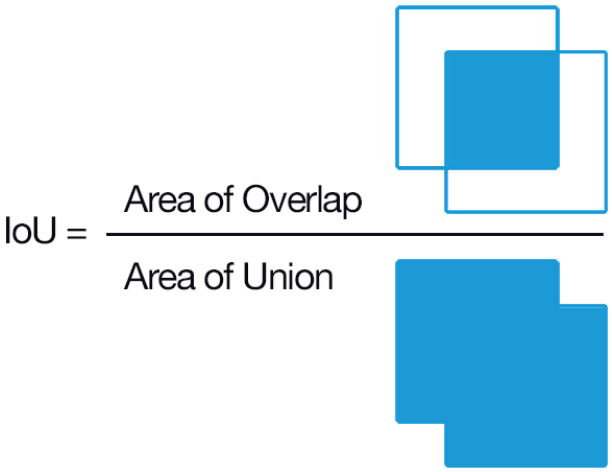
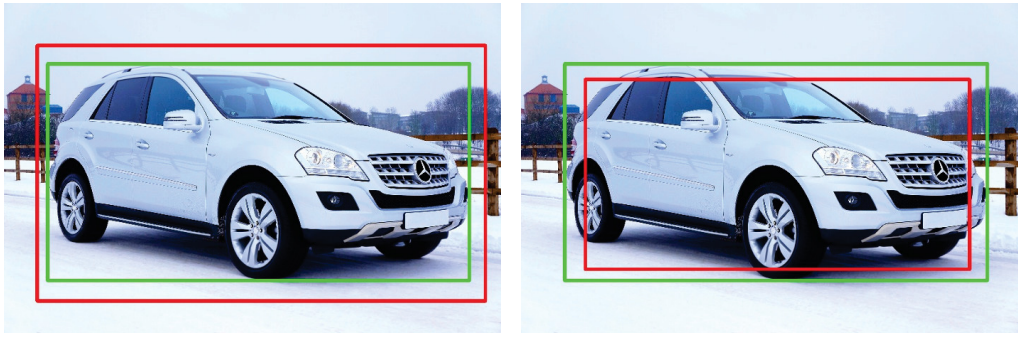


Figure 2.1: Definition of IoU. IoU is the division of the overlapping area between the bounding boxes by the union area. Image reproduced from `https://www.pyimagesearch.com/`.

The IoU metric is interesting because it penalizes both over- and under-estimated objects, as shown in Figure 2.2. Overestimated bounding boxes might include a large amount of unnecessary information and increase subsequent stages' processing time. On the other hand, meaningful parts of the object might be lost in underestimated bounding boxes.



(a) Overestimated bounding box (IoU = 0.8) (b) Underestimated bounding box (IoU = 0.8)

Figure 2.2: An illustration of two bounding boxes with the same IoU with the ground truth. The predicted position and ground truth are outlined in red and green, respectively. Image (without the bounding boxes) reproduced from `https://www.pexels.com`.

The PASCAL Visual Object Classes (VOC) (Everingham et al., 2010) and Common Objects in Context (COCO) (Lin et al., 2014b) object detection tasks considered a detection to be correct if the IoU between the predicted and ground-truth bounding boxes exceed 0.5. As stated by Everingham et al. (2010), this threshold was set deliberately low to account for inaccuracies

in bounding boxes in the training data, for example, defining the bounding box for a highly non-convex object (e.g., a person with arms and legs spread) is somewhat subjective.

Although LPs are convex objects, this threshold (IoU > 0.5) is by far the most adopted in the ALPR context because different datasets are labeled differently. For example, the bounding boxes of the LPs in the AOLP dataset (Hsu et al., 2013) are very tight (see Figure 2.3a), even cutting off parts of the LP characters in some cases, while other public datasets often consider the entire LP region as the bounding box. This is one of the reasons why some authors have re-labeled the bounding boxes in the AOLP dataset (see Figure 2.3b) to perform their experiments.

(a) Annotation provided by Hsu et al. (2013) (b) Annotation provided by Laroca et al. (2021b)

Figure 2.3: The way annotations are created differs considerably from dataset to dataset, as different authors follow different annotation protocols. (a) shows the original bounding box annotation for an LP from the AOLP dataset (Hsu et al., 2013), and (b) shows the bounding box annotation provided by Laroca et al. (2021b) for the same LP.

The ultimate goal of ALPR systems is to attain a high *recognition rate*, which is defined as the number of correctly recognized LPs divided by the number of LPs in the test set. Note that an LP is considered correctly recognized only if all its characters are accurately identified, as even a single misidentified character can lead to misidentification of the vehicle.

2.2 Deep Learning

Problems that are intellectually difficult for human beings but relatively straightforward for computers (e.g., problems that can be described by a list of mathematical rules) were rapidly tackled in the early days of Artificial Intelligence (AI). On the other hand, problems that humans solve intuitively, that feel automatic, such as telling the difference between pictures of cats and dogs, are very challenging for AI (Goodfellow et al., 2016; Redmon, 2018).

The ability to process natural data in their raw form (such as the pixel values of an image) was limited in conventional machine learning techniques. For many years, the development of machine learning systems required a lot of effort and considerable domain expertise to transform raw data into feature vectors with both discriminative and informative features (LeCun et al., 2015). It should be noted that the choice of data representation (or features) directly determines the performance of machine learning methods (Bengio et al., 2013), as demonstrated in Figure 2.4.

One solution to this problem is *representation learning*, which is a set of methods where the representations needed for detection or classification are automatically discovered from raw data (LeCun et al., 2015). In other words, instead of telling the system what a cat or dog looks like (through feature vectors), we provide as input a lot of images (i.e., millions or hundreds of thousands) of cats and dogs and let the system learns by itself to associate patterns and images with the correct label (Redmon, 2018). A string of empirical successes has been achieved both in academia and industry with the growing interest of the scientific community on representation learning (Bengio et al., 2013; LeCun et al., 2015; Bengio et al., 2021).

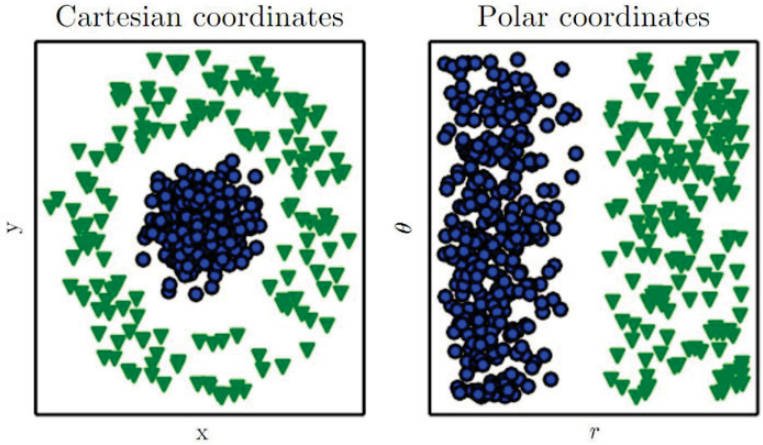


Figure 2.4: An example of different data representations. It is impossible to draw a straight line that separates two categories of data when representing them using Cartesian coordinates. On the other hand, this task becomes very simple when using Polar coordinates. Image reproduced from `http://www.deeplearningbook.org/`.

The central problem in representation learning is that it can be very difficult to extract such high-level, abstract features from raw data. *Deep learning* solves this problem by introducing representations that are expressed in terms of other, simpler representations (Goodfellow et al., 2016). An illustration of a deep learning model is shown in Figure 2.5. As can be seen, features regarding the presence or absence of edges at particular orientations and locations in the image are learned in the first representation layer. Next, corners and contours (i.e., collections of edges) are detected in the second layer. The third layer is where parts of objects are found by locating specific collections of contours and corners. Finally, the subsequent layers would detect specific objects as combinations of these parts (Goodfellow et al., 2016). The key aspect of deep learning is that these layers of features are learned from data using a general-purpose learning procedure, and thus it requires minimal engineering by hand (LeCun et al., 2015).

Initially, deep learning approaches were mainly employed for the handwritten digits recognition problem, breaking the supremacy of Support Vector Machines (SVMs) in the renowned MNIST dataset. The focus shifted progressively to object recognition in natural images, increasingly attracting the attention of the scientific community since the breakthrough achieved by Krizhevsky et al. (2012) on the ImageNet Large Scale Visual Recognition Challenge, bringing down the state-of-the-art error rate from 26.2% to 15.3% (Bengio et al., 2013).

In addition to the outstanding results achieved in several applications through deep learning, there are two other reasons for its success (Deng and Yu, 2014; LeCun et al., 2015; Bengio et al., 2021). First, the dramatically increased chip processing abilities (e.g., GPUs). Second, the fact that deep learning can easily take advantage of increases in the amount of available computation and data since it requires very little engineering by hand.

In the next two subsections, we provide more details about CNNs and GANs since they are two of the best known classes of deep neural networks and also those we explore in this work.

2.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs), also known as Convolutional Networks and ConvNets, are designed to process data that have a known, grid-like topology, for example, a color image composed of three 2-D arrays containing pixel intensities in the three color channels (LeCun et al., 2015; Goodfellow et al., 2016). It is worth noting that the impressive results reported by Krizhevsky et al. (2012) were obtained using CNNs.

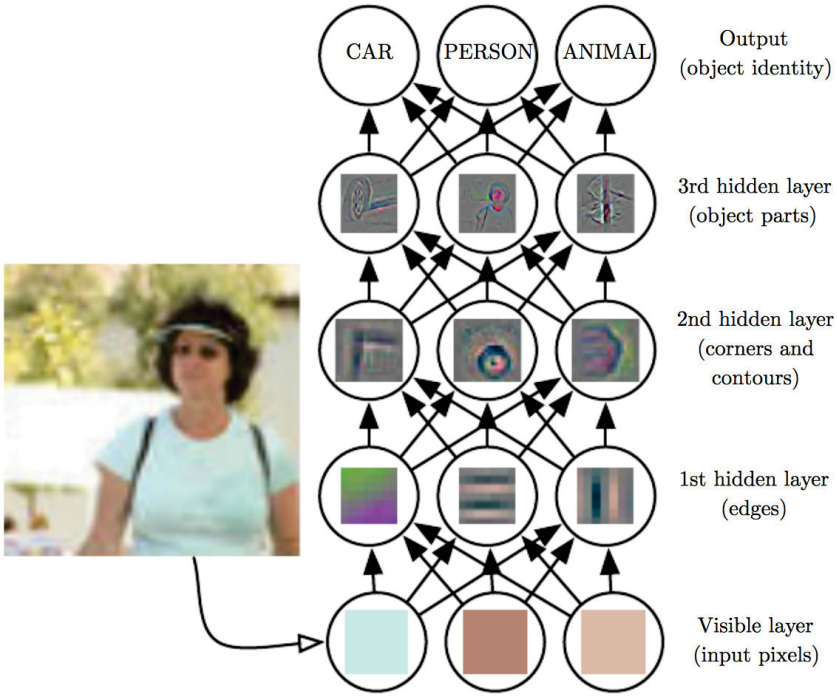


Figure 2.5: An illustration of a deep learning model. First, low-level features such as edges and curves are found, and then more abstract concepts are built through a series of layers. Image reproduced from `http://www.deeplearningbook.org/`.

All CNNs perform a kind of linear operation called *convolution* (hence the name) in at least one of their layers (Goodfellow et al., 2016). The basic building blocks of CNNs are convolutions, pooling (downsampling) operators, activation functions (e.g., Rectified Linear Unit (ReLU)) and fully connected layers, which are essentially similar to hidden layers of a Multilayer Perceptron (MLP) (Ponti et al., 2017). Each one of those building blocks will be described throughout this section. Figure 2.6 shows an example of a CNN.

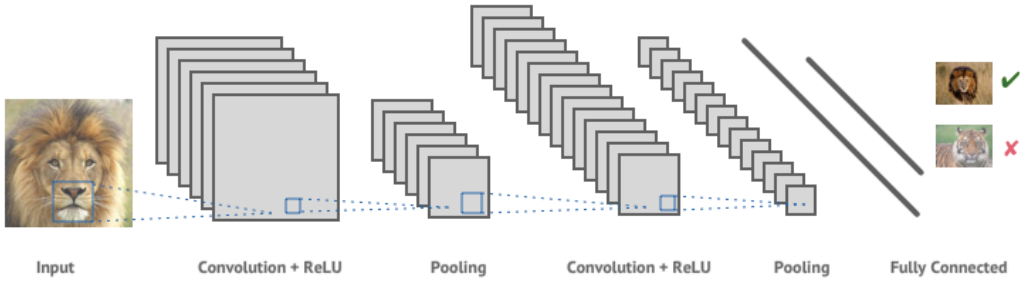


Figure 2.6: An example of a CNN, which consists of convolutional layers, activation functions and pooling layers, followed by a set of fully connected layers. Image reproduced from (Tejani, 2016).

2.2.1.1 Convolutional Layer

The main building blocks of CNNs are the convolutional layers, which are composed of a set of *filters* (or kernels), each to be applied to the entire array of pixel values. Each filter is a matrix of weights (or values) that can be considered as a feature identifier (e.g., straight edges, simple colors, and curves). The filters produce what can be seen as an affine transformation of the input image (Ponti et al., 2017). Each filter is slid (or convolved) around the input image, with the

values in the filter being multiplied by the original pixel values of the image (Ponti et al., 2017). An example of 2-D convolution is shown in Figure 2.7.

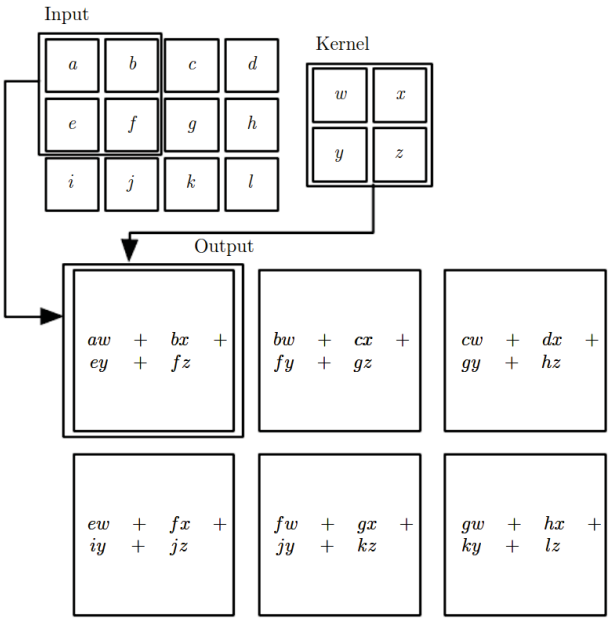


Figure 2.7: An example of 2-D convolution. The boxes with arrows were drawn to indicate how the upper-left element of the output tensor is formed by applying the kernel to the corresponding upper-left region of the input tensor. Image reproduced from (Goodfellow et al., 2016).

Each region that the filter processes is called a *local receptive field*, and an output value (pixel) is a combination of the input pixels in this local receptive field, as shown in Figure 2.8. That makes the convolutional layer different from layers of an MLP, where each neuron produces a single output based on all values from the previous layer (see Figure 2.9) (Ponti et al., 2017).

An important aspect of CNNs is that the filter weights are shared across receptive fields, significantly reducing the number of weights that the network has to learn. As stated by LeCun et al. (2015), if a feature can appear in one part of the image, it could appear anywhere, hence the idea of filters at different locations sharing the same weights and detecting the same pattern in different parts of the array.

Note that convolution is not naturally equivariant to some other transformations, such as changes in the scale or rotation of an image. Therefore, other mechanisms are necessary for handling these kinds of transformations (Goodfellow et al., 2016).

2.2.1.2 Activation Function

In order to go from one layer to the next, a set of units compute a weighted sum of their inputs from the previous layer and pass the result through an activation function (LeCun et al., 2015). In contrast to using a sigmoid function such as the logistic or hyperbolic tangent in MLPs, the *Rectified Linear Unit* (ReLU) is often used in CNNs after convolutional or fully connected layers (Ponti et al., 2017). Figure 2.10 shows plots of these functions.

Although sigmoid functions are commonly used in neural networks, their limitations are well known. For example, it is slow to learn the whole network due to weak gradients when the units are close to saturation in both directions (Deng and Yu, 2014). Deep CNNs with ReLUs train several times faster than their equivalents with sigmoid functions (Krizhevsky et al., 2012). The *Leaky* ReLU allows for a small, non-zero gradient when the unit is saturated and inactive.

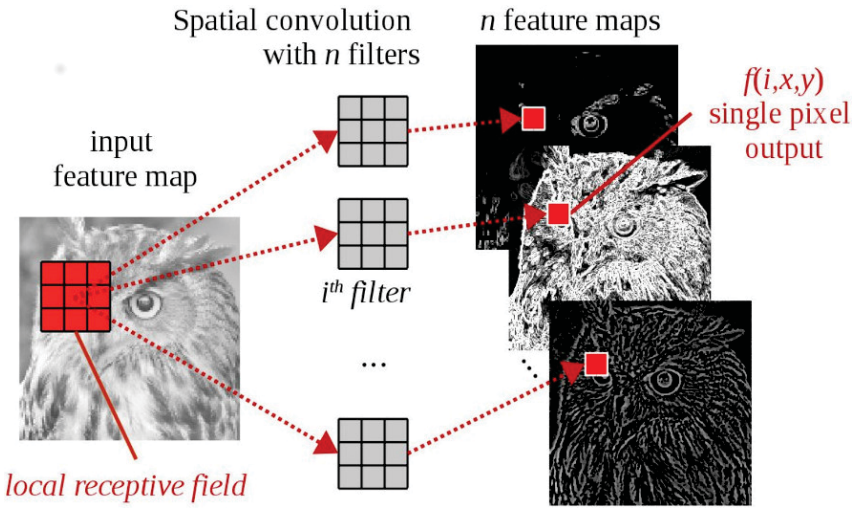
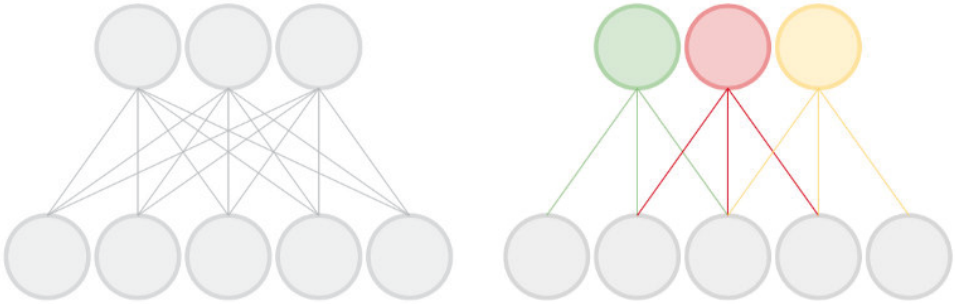
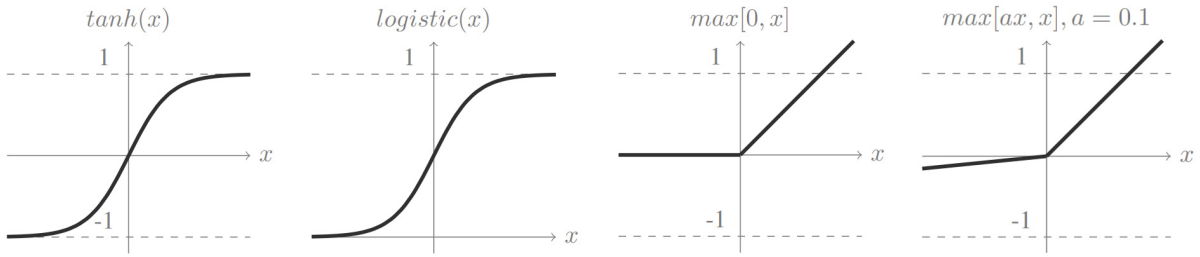


Figure 2.8: The convolution process. It processes local information centered in each position (x, y): this region is a called local receptive field, whose values are used as input by some filter i with weights w_i in order to produce a single point (pixel) in the output feature map $f(i, x, y)$. Image reproduced from (Ponti et al., 2017).



(a) Fully connected layer (b) Convolutional layer

Figure 2.9: Comparison between fully connected (a) and convolutional layers (b). In a fully connected layer, each unit is connected to all units of the previous layers. On the other hand, in a convolutional layer, each unit is connected to a constant number of units in a local region of the previous layer. Image reproduced from `https://www.quora.com/what-is-the-difference-between-a-convolutional-neural-network-and-a-multilayer-perceptron`.



(a) Hyperbolic tangent (b) Logistic (c) ReLU (d) Parametric ReLU (PReLU)

Figure 2.10: Activation functions. (a) and (b) are often used in MLP networks, while (c) and (d) are more common in CNNs. A PReLU (d) with $a = 0.01$ is equivalent to Leaky ReLU. Image reproduced from (Ponti et al., 2017).

Maas et al. (2013) observed that the non-zero gradient does not substantially affect training optimization and that deep networks with Leaky ReLUs converge slightly faster.

In addition to the innovations in better architectures of deep learning models, there is also a growing body of work on developing and implementing better nonlinear units (Ramachandran et al., 2018; Misra, 2020; Nader and Azar, 2020).

2.2.1.3 *Pooling*

In addition to convolutions and activation functions, *pooling* operations make up another important building block in CNNs. Pooling operations reduce the size of feature maps by using some function to summarize subregions, such as taking the average or the maximum value of the contributing features (Dumoulin and Visin, 2018). Although much better linear discrimination performance was achieved with max pooling compared to average pooling in (Boureau et al., 2010a), the same research group showed in (Boureau et al., 2010b) that depending on the data and features, either max or average pooling may perform best. Then, in this section, we focus on the max-pooling operator since it is the most frequently used (Ponti et al., 2017).

The role of the pooling layer is to merge semantically similar features into one, enabling representations to vary very little when elements in the previous layer vary in position and appearance (LeCun et al., 2015). In other words, the use of pooling can be viewed as adding an infinitely strong prior that the function the layer learns must be invariant to small translations (Goodfellow et al., 2016). See Figure 2.11 for an example of how max pooling works.

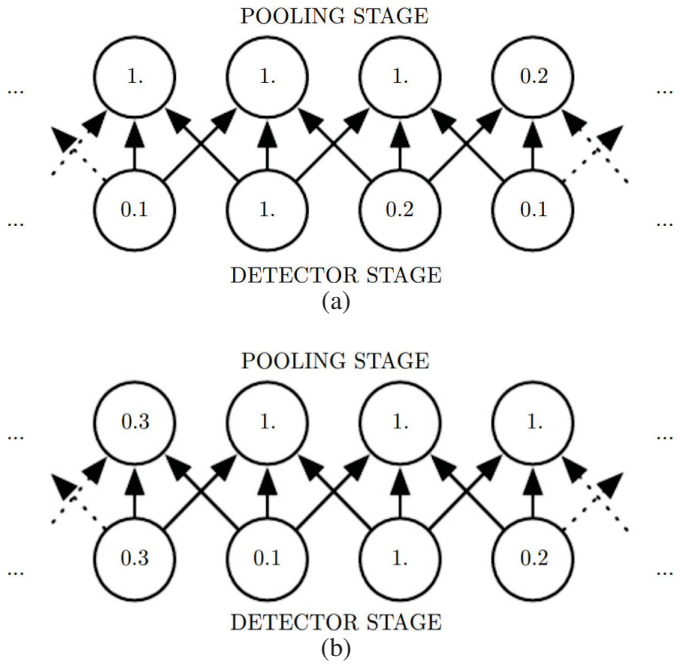


Figure 2.11: Max pooling introduces invariance. (a) shows a view of the middle of the output of a convolutional layer, and (b) shows a view of the same network after the input has been shifted to the right by one pixel. The bottom row shows the outputs of the activation function. The top row shows the outputs of max pooling, with a stride of one pixel between pooling regions and a pooling region width of three pixels. Observe that every value in the bottom row has changed, but only half of the values in the top row have changed. This occurred because the max-pooling units are only sensitive to the maximum value in the neighborhood, not its exact location. Image reproduced from `http://www.deeplearningbook.org/`.

It is possible to use fewer pooling units than detector units (see Figure 2.12), as pooling summarizes the responses over a whole neighborhood. In this way, the computational efficiency of the network is improved because the next layer has fewer inputs to process. When the number of parameters in the next layer is a function of its input size (e.g., the next layer is fully connected and based on matrix multiplication), this reduction in the input size can also result in improved statistical efficiency and reduced memory requirements for storing the parameters (Goodfellow et al., 2016).

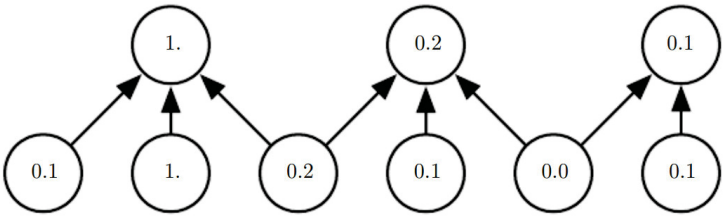


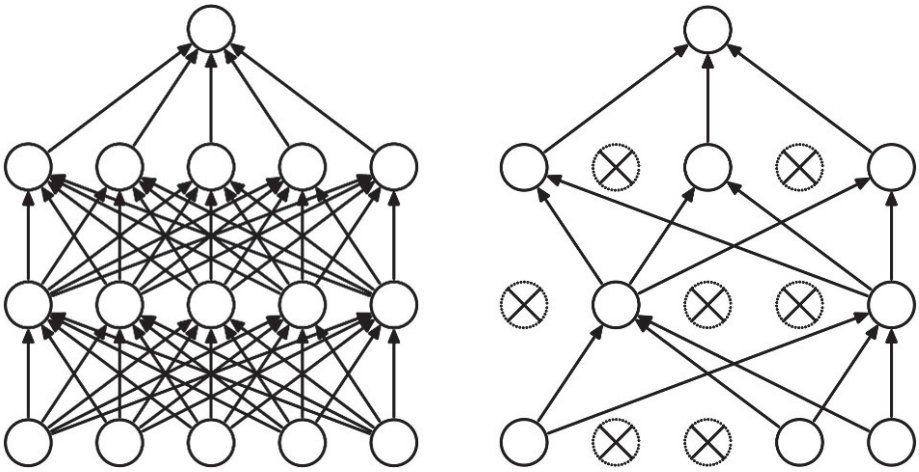
Figure 2.12: Max pooling with downsampling. When using stride = 2 between pools, the representation size is reduced by a factor of two, which reduces the computational and statistical burden on the next layer. Note that the rightmost pooling region is smaller but must be included if we do not want to ignore some of the detector units. Image reproduced from `http://www.deeplearningbook.org/`.

It should be noted that generative models such as auto-encoders and GANs shown to be harder to train with pooling layers (Radford et al., 2016; Ponti et al., 2017). Therefore, pooling layers might be avoided in some neural network architectures.

2.2.1.4 Fully Connected Layers and Regularization

Conventional CNNs perform convolution in the lower layers of the network. For classification, the feature maps of the last convolutional layer are vectorized and fed into fully connected layers followed by a softmax logistic regression layer (Lin et al., 2014a).

However, the fully connected layers are prone to overfitting, thus hampering the generalization ability of the overall network (Lin et al., 2014a). In this sense, a technique called *dropout* (Srivastava et al., 2014) was introduced to limit co-adaptation. It operates as follows. On each training instance, each hidden unit is randomly omitted with a fixed probability (e.g., $p = 0.5$) (Deng and Yu, 2014). The neurons that are "dropped out" do not contribute to the forward pass and do not participate in backpropagation, as illustrated in Figure 2.13. Thus, the neural network samples a different architecture every time an input is presented, but all these architectures share weights (Krizhevsky et al., 2012).



(a) Standard neural network (b) After applying dropout

Figure 2.13: An illustration of dropout regularization. (a) shows a standard neural network with two hidden layers, and (b) shows an example of a thinned network produced by applying dropout to the network on (a). Image reproduced from (Srivastava et al., 2014).

Dropout is turned off in the test stage, and the activations are rescaled by p to compensate those activations that were dropped during the training stage (Ponti et al., 2017). The benefits of

dropout regularization for training deep neural networks are to make a hidden unit act strongly by itself without relying on others and to serve as a way to do model averaging of different networks. These benefits are most pronounced when the training data is limited or when the network size is disproportionally large with respect to the size of the training data (Deng and Yu, 2014).

Deep neural networks involve the composition of several functions or layers. Training them is complicated because the distribution of each layer's inputs changes during training, as the parameters of the previous layers change (Ioffe and Szegedy, 2015). In other words, the gradient tells how to update each parameter under the assumption that the other layers do not change. In practice, all layers are updated simultaneously. Hence, unexpected results might happen because many functions composed together were changed simultaneously, using updates computed under the assumption that the other functions would remain constant (Goodfellow et al., 2016).

This makes it notoriously hard to train models with saturating nonlinearities. Therefore, the training is slower since it requires lower learning rates and careful parameter initialization (Ioffe and Szegedy, 2015). In this direction, Ioffe and Szegedy (2015) proposed a regularization technique called *batch normalization* for controlling the distributions of neural network activations, thereby reducing internal covariate shift (Cooijmans et al., 2017). Batch normalization is a method of adaptive reparametrization in which the output of each neuron (before application of the nonlinearity) is normalized by the mean and standard deviation of the outputs calculated over the examples in the mini-batch (Salimans and Kingma, 2016). This effectively decouples each layer's parameters from those of other layers, leading to a better-conditioned optimization problem. Deep neural networks trained with batch normalization converge significantly faster, generalize better, and often do not need dropout (Cooijmans et al., 2017; Ponti et al., 2017).

2.2.2 Generative Adversarial Networks (GANs)

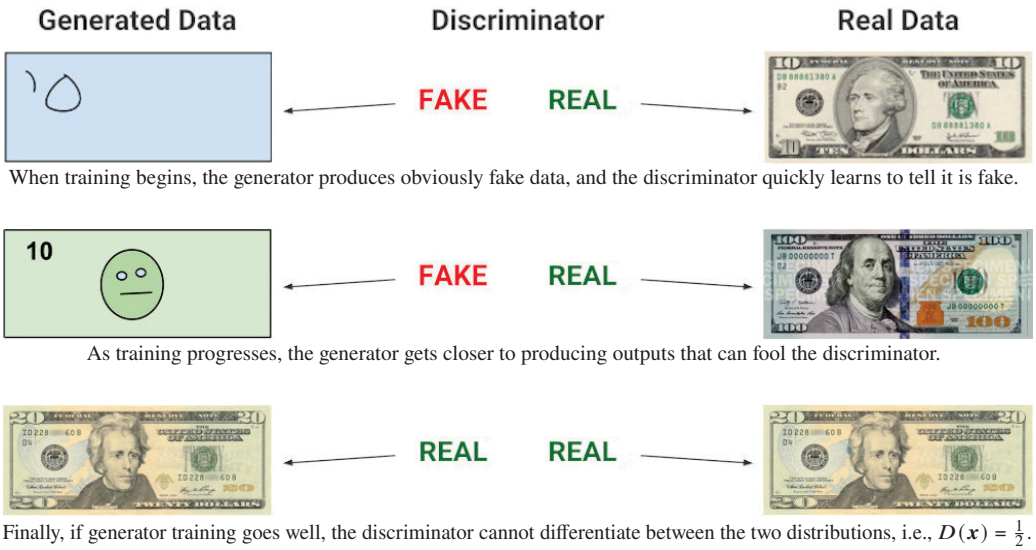
Compared with discriminative models, which only model the decision boundary between the classes, generative models tackle a more difficult task: to capture the actual distribution of each class in order to generate similar data (Oussidi and Elhassouny, 2018; Harshvardhan et al., 2020). In other words, as defined by Goodfellow (2016), generative models refer to any model that takes a training set, consisting of samples drawn from a distribution p_{data}, and learns to represent an estimate of that distribution somehow. The result is a probability distribution p_{model}.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014b) are generative models based on a competition between a *generator* network G and a *discriminator* network D. The generator $G(\boldsymbol{z}; \theta^{(G)})$ produces samples from the data distribution, $p_{\text{data}}(\boldsymbol{x})$, by transforming vectors of noise $\boldsymbol{z}$ as $\boldsymbol{x} = G(\boldsymbol{z}; \theta^{(G)})$ (Goodfellow et al., 2014b). The function G is simply a function represented by a neural network that transforms the random, unstructured $\boldsymbol{z}$ vector into structured data, intended to be statistically indistinguishable from the training data. The training signal for G is provided by the discriminator network $D(\boldsymbol{x})$, which is trained to distinguish samples from the generator distribution $p_{\text{model}}(\boldsymbol{x})$ from real data. In turn, the generator network G is trained to fool the discriminator into accepting its outputs as being real (Salimans et al., 2016). At convergence (a local Nash equilibrium), the generator's samples are indistinguishable from real data ($p_{model} = p_{data}$), and the discriminator outputs $1/2$ everywhere[3] (Fedus et al., 2018; Harshvardhan et al., 2020). Therefore, neither player can improve its payoff, and the discriminator may then be discarded (Goodfellow et al., 2016).

Goodfellow et al. (2014b) observed that the generator can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the

[3] This ($p_{model} = p_{data}$) is just an example of an idealized case; generally, the generator does not need to produce perfect replicas from the input domain to be useful (Brownlee, 2019; Goodfellow, 2019).

discriminator is analogous to the police, trying to detect the counterfeit currency. Competition between counterfeiters and police leads to more and more realistic counterfeit money, until eventually the counterfeiters produce perfect fakes and the police cannot distinguish between real and fake money. Figure 2.14 illustrates this alternate training process.



When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell it is fake.

As training progresses, the generator gets closer to producing outputs that can fool the discriminator.

Finally, if generator training goes well, the discriminator cannot differentiate between the two distributions, i.e., $D(\boldsymbol{x}) = \frac{1}{2}$.

Figure 2.14: An illustration of the fundamental intuition underlying the training process of GANs. Image adapted from `https://developers.google.com/machine-learning/gan/gan_structure`.

GANs typically use CNNs as the generator and discriminator models (Brownlee, 2019). The most common training algorithm is simply to use a gradient-based optimizer to repeatedly take simultaneous steps on both players, incrementally minimizing each player's cost with respect to that player's parameters. In simpler terms, the back-propagation algorithm propagates gradients from the discriminator through the generator's output (Goodfellow et al., 2020). The Adam optimizer (Kingma and Ba, 2015) has been chosen in most works in the literature (Miyato et al., 2018; Lučić et al., 2019; Choi et al., 2020; Wang et al., 2021b). At the end of the training process, GANs can often produce realistic samples, as shown in Figure 2.15.

Figure 2.15: These images are samples from StyleGAN2 (Karras et al., 2020) depicting three people who do not exist but were "imagined" by a GAN after training on a high-quality image dataset of human faces. The three images were downloaded from `https://thispersondoesnotexist.com/`.

Indeed, GANs are often regarded as producing the best samples compared to other generative models, such as Variational Autoencoders (VAEs), especially in generating realistic high-resolution images (Goodfellow et al., 2016; Wang et al., 2018b; Karras et al., 2020). Note that they have proven useful for several tasks other than straightforward image generation (Goodfellow et al., 2020). Consequently, they have become a hot research topic. According to Gui et al.

(2023), approximately 28,500 GAN-related papers were published in 2020 alone, constituting approximately 78 papers every day or more than three per hour.

However, GANs are not without problems. The two most significant are that they are hard to train and difficult to evaluate (Salimans et al., 2016; Wang et al., 2021b). Regarding being difficult to train, Odena et al. (2018) stated that various causes seem to plague GANs' training procedure. The most notable of them, called *mode collapse*, is characterized by a tendency of the generator to output samples from a small subset of the modes of the data distribution. In extreme cases, the generator outputs only a few unique samples or even just the same sample repeatedly. As even the best learning algorithms often fail to converge (Goodfellow et al., 2020), several works have sought to design better costs, models, and training algorithms with better convergence properties (Arjovsky et al., 2017; Miyato et al., 2018; Bang and Shim, 2021). In terms of evaluation, generative models are traditionally evaluated in terms of fidelity (how realistic a generated image is) and diversity (how well generated samples capture the variations in real data) of the learned distribution (Borji, 2022). Nevertheless, there is not a single compelling way to evaluate both fidelity and diversity simultaneously (Goodfellow, 2016). The two most common GAN evaluation measures are Inception Score (IS) (Salimans et al., 2016) and Fréchet Inception Distance (FID) (Heusel et al., 2017), which rely on pre-trained deep networks to represent and statistically compare original and generated samples (Borji, 2022). However, several shortcomings of both measures have been pointed out over the years (Shane Barratt, 2018; Shmelkov et al., 2018; Borji, 2022). That is why some authors (Theis et al., 2016; Borji, 2022) argued that generative models, including GANs, need to be evaluated with respect to the application(s) they are intended for (evaluation metrics should be tailored to the target task).

Considering the success achieved by GANs in recent years, there are many relevant derivatives of GANs proposed in the literature. In the following subsections, we review two of them given their importance and because we have explored them in the development of this work.

2.2.2.1 Deep Convolutional Generative Adversarial Networks (DCGANs)

The original GANs (Goodfellow et al., 2014b) worked but were unstable and difficult to train, especially with large inputs, often resulting in generators that produce nonsensical outputs. Nevertheless, shortly afterward, Radford et al. (2016) crafted a *Deep Convolutional Generative Adversarial Network* (DCGAN)[4] that showed stable training across a range of datasets and allowed for training higher resolution and deeper generative models. Based on this, most GANs proposed after (Radford et al., 2016) are at least loosely based on the DCGAN architecture (Goodfellow, 2016; Wang et al., 2021b; Gui et al., 2023).

DCGANs have three main differences from the original GANs: (i) DCGAN replaces any pooling layers with strided convolutions (see the generator used by Radford et al. (2016) for scene modeling in Figure 2.16), allowing each network to learn its own spatial downsampling; (ii) DCGAN uses batch normalization in most layers of both the discriminator and the generator (except for the G output layer and D input layer to avoid sample oscillation and model instability) to deal with training problems that arise due to poor initialization, preventing mode collapse; and (iii) DCGAN uses ReLU in G for all layers except for the output, and Leaky ReLU for all layers in D – while ReLU allowed the model to learn quicker how to saturate and cover the color space of the training distribution, Leaky ReLU worked well for higher resolution modeling.

[4] Although GANs were both deep and convolutional prior to DCGANs, the name DCGAN is traditionally used to refer to this specific style of architecture (Goodfellow, 2016).

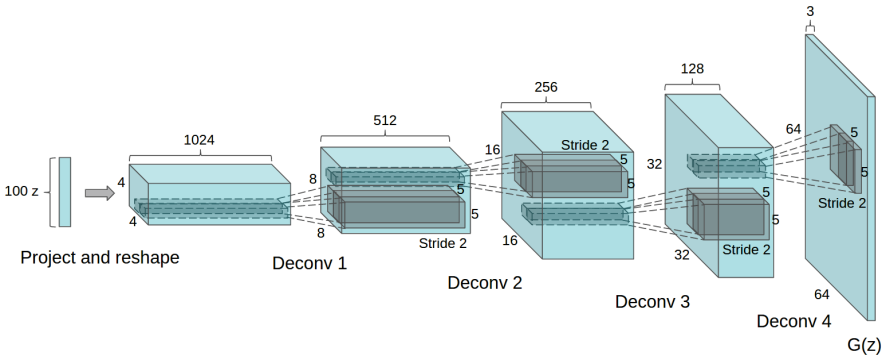


Figure 2.16: The generator of DCGAN with four sequential fractionally-strided convolutional layers, which convert a 100-dimensional uniform distribution z – projected to a small spatial extent convolutional representation with many feature maps – into a 64×64 image. Image reproduced from (Radford et al., 2016).

2.2.2.2 *Conditional Generative Adversarial Networks* (cGANs)

Although standard (or unconditioned) GAN models are able to generate new random plausible examples for a given dataset, there is no way to control the appearance (e.g., class) of the samples that are generated other than trying to figure out the complex relationship between the latent space input to the generator and the generated images (Kaneko et al., 2017; Brownlee, 2019). With that in mind, Mirza and Osindero (2014) proposed to extend GANs to a conditional model – called *Conditional Generative Adversarial Network* (cGAN) – by conditioning both the generator and discriminator on some extra label y, which can be any kind of auxiliary information such as class labels or data from other modalities. Figure 2.17 compares GANs and cGANs in a simplified way.

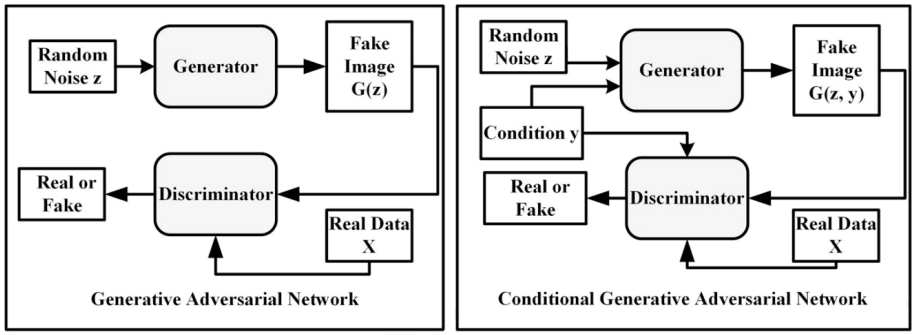


Figure 2.17: Comparison between GANs and cGANs. Image reproduced from (Cheng et al., 2020).

Over the years, many studies have empirically shown that there is almost always a causal relationship between using labels in any way, shape or form and a dramatic improvement in the subjective quality of the samples generated by GAN models (Denton et al., 2015; Salimans et al., 2016; Odena et al., 2018), even though it is not entirely clear why this trick works in each specific case (Goodfellow, 2016; Boulahbal et al., 2021). An important characteristic of cGAN models is that the generated images should not only be realistic but also recognizable as related to the specified condition y (e.g., coming from a given class) (Shmelkov et al., 2018).

In practical terms, cGANs are trained on a labeled dataset, allowing the label for each generated instance to be specified. cGANs find applications in several areas such as categorical image generation using class labels (Mirza and Osindero, 2014; Miyato and Koyama, 2018), text-to-image synthesis, where text sentences are converted into images (Reed et al., 2016; Zhang et al., 2021b), and image-to-image translation, where one image is transformed into another (Isola et al., 2017; Zhu et al., 2017a). In the subsequent paragraphs, we elaborate on image-to-image translation, as we plan to use cGANs to generate LP images from LP masks.

Image-to-image translation (see Figure 2.18) is a class of problems where the goal is to translate images from one domain to another by learning a mapping between the input and output images using a training dataset of *paired* (Isola et al., 2017; Shaham et al., 2021) or *unpaired* (Zhu et al., 2017b; Lee et al., 2020) cross-domain image pairs. It should be noted that even though the latter approach (*unpaired*) is generally called unsupervised as a counterpart of the former, it actually assumes that the domain labels are given *a priori* (Baek et al., 2021b).

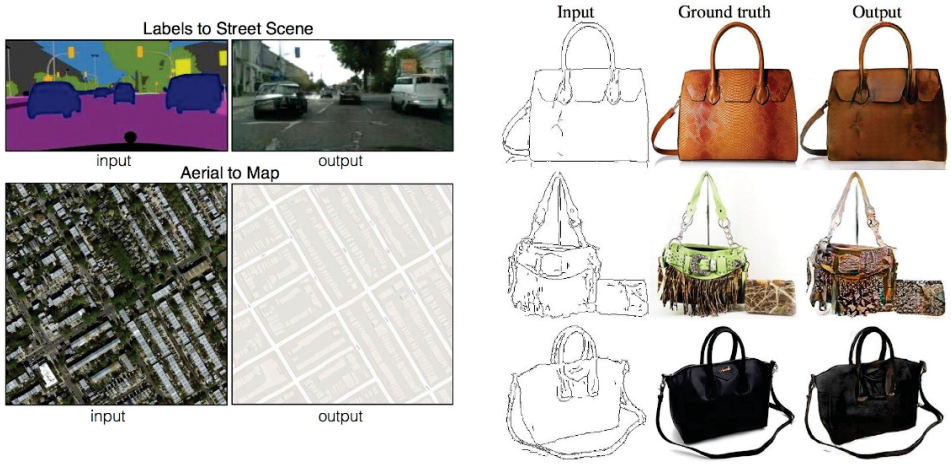


Figure 2.18: Image-to-image translation is a concept introduced by Isola et al. (2017) that encompasses many kinds of transformations of an image: converting segmentation masks into images, converting aerial photos into maps, converting sketches into photorealistic images, among others. Image adapted from (Isola et al., 2017).

Figure 2.19 illustrates the difference between paired and unpaired training data in image-to-image translation. The application of cGANs to this task was first investigated by Isola et al. (2017), who created a model – called *pix2pix* – that maps an image from input to output domain using an adversarial loss in conjunction with the L1 loss between the result and target, thus requiring paired training data. Since this seminal work, paired image-to-image translation models have shown impressive results (Wang et al., 2018b; Park et al., 2019; Shaham et al., 2021; Zhou et al., 2021). Nevertheless, acquiring such training data (i.e., matching image pairs with pixelwise or patchwise labeling) can be time-consuming and even unrealistic (Zhu et al., 2017a; Lee et al., 2020). For example, for converting daylight scenes to night scenes and vice versa, even though matching image pairs can be obtained with stationary cameras, moving objects in the scene (e.g., vehicles and clouds) often cause varying degrees of content discrepancies (Yi et al., 2017). To tackle this problem, CycleGAN (Zhu et al., 2017b), DualGAN (Yi et al., 2017) and DiscoGAN (Kim et al., 2017) provided a new insight (nearly at the same time), in which the GAN models discover relations between two visual domains without any explicitly paired data. As paired data is often not available, unpaired image-to-image translation has gained a great deal of attention in recent years (Zhao et al., 2020b; Tang et al., 2021; Zheng et al., 2021).

As a side note, with paired training data, image-to-image translation can be approached by a single feedforward CNN trained to minimize a regression loss (Chen and Koltun, 2017). However, as stated by Goodfellow (2016), models with generative modeling are better trained for this task because there are multiple correct outputs for each input (as shown in Figure 2.20).

2.3 Data Augmentation

A huge number of training examples are required to train deep networks since they often have a large set of parameters to be optimized (Ponti et al., 2017; Bengio et al., 2021). In practice,

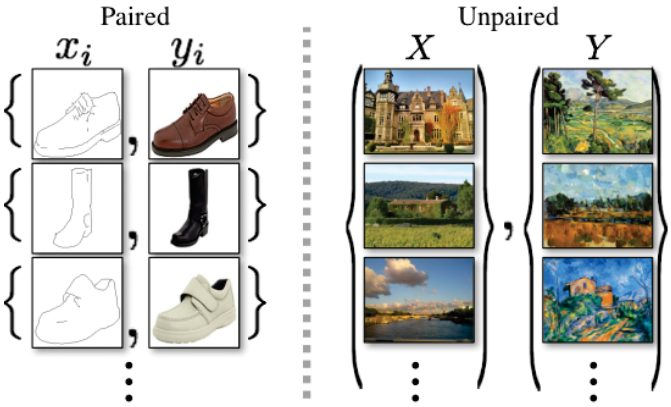


Figure 2.19: *Paired* training data (left) consists of training examples $\{x_i, y_i\}_{i=1}^{N}$, where the correspondence between x_i and y_i exists. *Unpaired* training data (right) consists of a source set $\{x_i\}_{i=1}^{N}$ ($x_i \in X$) and a target set $\{y_j\}_{j=1}^{M}$ ($y_j \in Y$), with no information provided as to which x_i matches which y_j. Image reproduced from (Zhu et al., 2017b).

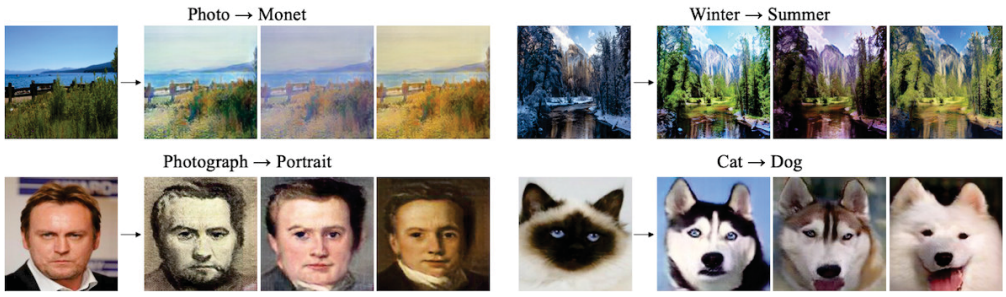


Figure 2.20: Examples of diverse outputs produced by DRIT++ (Lee et al., 2020) trained without aligned pairs. Observe that a single input may correspond to multiple possible outputs. Image adapted from (Lee et al., 2020).

the amount of data available is limited. One way to get around this problem is to create fake data and add it to the training set. This process is known as *data augmentation*. It is reasonably straightforward to create new fake data for some machine learning tasks (Goodfellow et al., 2016).

Images in the same dataset usually have similar illumination conditions, a low variance of rotation, pose, etc. Therefore, one can augment the training dataset using many operations to produce several times more examples (Ponti et al., 2017). To better illustrate, Figure 2.21 shows multiple images created from a single one using Albumentations (Buslaev et al., 2020), which is a well-known library for image augmentation. Operations like translating the training images a few pixels in each direction can often greatly improve generalization, even if the model has already been designed to be partially translation-invariant by using the convolution and pooling techniques described in the previous section. Many other operations, such as rotating or scaling the image, have also proven quite effective (Goodfellow et al., 2016; Ponti et al., 2017).

It is well-known that unbalanced data (usually the case in ALPR) is undesirable for neural network classifiers since the learning of some patterns might be biased. This problem can be addressed with data augmentation, by increasing the number of images of under-represented classes to create a new set of training images, in which each class is equally represented.

It is worth noting that some frameworks already have built-in data augmentation (Redmon et al., 2016), and one must be careful not to apply transformations that would change the correct class. For example, OCR tasks require recognizing the difference between ‘b’ and ‘d’ and the difference between ‘6’ and ‘9’, so these cases must be considered before applying horizontal flips and 180° rotations for those tasks (Laroca et al., 2018; Aberdam et al., 2021).

Figure 2.21: An example of how some augmentations can be applied to create new images from the original one. Image reproduced from `https://github.com/albumentations-team/albumentations/`.

3. RELATED WORK

This chapter reviews relevant works that explored deep learning methodologies within the ALPR domain. For a broader survey covering research on traditional image processing techniques, please refer to (Du et al., 2013; Lubna et al., 2021; Shashirangana et al., 2021).

We structured this chapter into five sections. The first two sections focus on methods designed or adjusted for LPD and LPR. The third section explores research that leverages data synthesis techniques to enhance the performance of LPR models. The fourth section provides a brief overview of approaches that do not align with the preceding sections, such as models for locating the four corners of the LPs, commercial ALPR systems, and established methods for scene text recognition. The final section offers concluding remarks.

3.1 License Plate Detection (LPD)

Many authors have addressed the LPD stage using off-the-shelf object detection CNNs. Considering that there is a large portion of works on ALPR that are focused on the recognition stage, many authors simply employed well-known detectors without providing details about the implementation, training strategies, and results obtained. For example, Zhang et al. (2018a) used Faster-RCNN (Ren et al., 2017), Zhang et al. (2019b, 2021c); Kim et al. (2021) explored YOLOv2 (Redmon and Farhadi, 2017), and Zhang et al. (2021d) used YOLOv4 (Bochkovskiy et al., 2020) for LPD. The following paragraphs describe relevant works where more information was provided regarding the methods used/designed for the detection stage.

Henry et al. (2020) employed Fast-YOLOv3 for locating the LPs directly in the input image (i.e., without vehicle detection)[5]. Although high precision and recall rates were achieved in five different datasets, the chosen datasets were collected under relatively controlled conditions (e.g., with handheld cameras in parking lots or stationary cameras in car wash facilities) and the authors trained a distinct network for each dataset, i.e., the parameters (e.g., network input size) were adjusted specifically for each scenario. In this way, it is not clear whether such a shallow network (compared to state-of-the-art object detectors) is robust enough to handle multiple real-world scenarios. Silva and Jung (2017, 2020), on the other hand, noticed that the Fast-YOLO model achieved a low recall rate when detecting LPs without prior vehicle detection. Therefore, they used the Fast-YOLO model arranged in a cascaded manner to first detect the frontal view of the cars and then locate their LPs in the detected patches, attaining high precision and recall rates. Their approach can remarkably process 185 frames per second (FPS) on an NVIDIA TITAN X GPU, assuming that a single vehicle is being processed.

Inspired by this cascaded approach, Laroca et al. (2018) first fine-tuned the YOLOv2 model (Redmon and Farhadi, 2017) to locate the vehicles (both front and rear views) in the input image and then trained the Fast-YOLOv2 model to detect the respective LPs in the cropped patches. The authors reported promising speed/accuracy results in two public datasets acquired in Brazil. An important finding of their work is that better results were reached when using two distinct classes for detecting cars and motorcycles (instead of a single class called "vehicle"). On the other hand, Silva and Jung (2018) detected the vehicles in the input image using the pre-trained

[5] Each YOLO model has a corresponding smaller version known as YOLO-tiny (or Fast-YOLO). These variants have fewer convolutional layers and filters than their larger counterparts. Despite their compact design, YOLO-tiny versions can still achieve a surprising level of detection accuracy (Redmon et al., 2016), leading to their adoption in various real-world applications (Bezerra et al., 2018; Salomon et al., 2020; Ismail et al., 2021; Ke et al., 2023).

YOLOv2 model (i.e., without any change or refinement). The outputs related to vehicles (i.e., cars and buses) were merged, whereas those related to other classes were ignored. Then, they proposed a Warped Planar Object Detection Network (WPOD-NET) that searches for LPs and regresses one affine transformation per detection, enabling a rectification of the LP region to a rectangle resembling a frontal view. Their approach, illustrated in Figure 3.1, was trained using many synthetically warped versions of real images to augment the training dataset composed of less than 200 manually labeled images. The detection stage's results and execution time were not reported, as the authors focused on the end-to-end evaluation of their system.

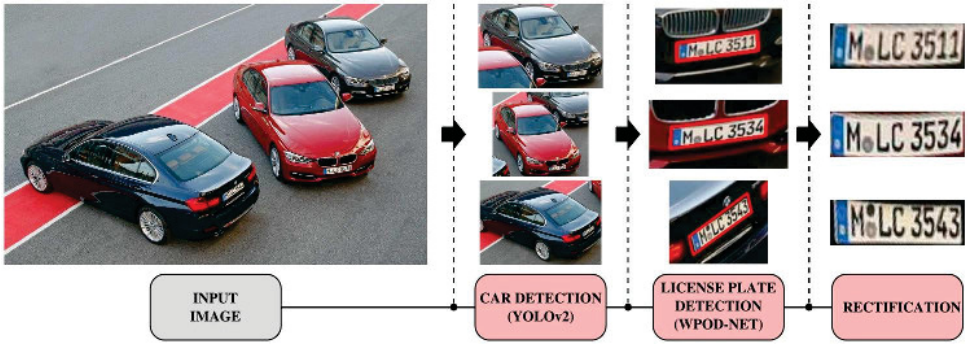


Figure 3.1: The LPD approach proposed by Silva and Jung (2018). Note that the rectification process can significantly help the OCR task when the LPs are heavily distorted. Image reproduced from (Silva and Jung, 2018).

Considering some limitations of WPOD-NET, such as not working properly for motorcycle LPs due to differences in aspect ratio and layout, Silva and Jung (2022) presented an Improved Warped Planar Object Detection Network (IWPOD-NET) that learns separately the weights for the classification and localization tasks. In summary, while WPOD-NET relies on weight sharing for both tasks until the last layer, IWPOD-NET contains two shallow (but independent) sub-networks, one for each task. By massively exploring data augmentation techniques and post-processing strategies, IWPOD-NET reached remarkable performance for handling both car and motorcycle LPs captured at a variety of lighting conditions and viewpoints.

Xie et al. (2018) proposed a YOLO-based model to predict the LP rotation angle in addition to its coordinates and confidence value. Their network consists of seven convolutional layers and three fully connected ones. Before that, another CNN (with the same architecture) was applied to determine the attention region in the input image, assuming that some distance will inevitably exist between any two LPs. By cascading both models, their approach outperformed all baselines in three public datasets while still running in real time. Despite the impressive results, it is important to highlight two limitations in their work: (i) the authors simplified the problem by forcing their ALPR system to output only one bounding box per image – this limitation was also highlighted by Zhang et al. (2021a); and (ii) motorcycle LPs might be lost when determining the attention region since, in some scenarios (e.g., traffic lights), they might be very close.

Rather than exploring off-the-shelf object detectors, Li et al. (2018) trained a 4-layer CNN using characters cropped from general text to perform a character-based LP detection. The network was employed in a sliding-window fashion across the entire image to generate a text salience map. Text-like regions were extracted based on the clustering nature of the characters. Connected Component Analysis (CCA) was subsequently applied to produce the initial candidate boxes. Then, an LP/non-LP CNN – also with four layers – was trained to remove false positives. Finally, the bounding boxes were refined through a projection-based method. Although the precision and recall rates obtained were higher than those achieved in previous works, this sequence of methods (see Figure 3.2) is too expensive for real-time applications, taking more than 2 seconds to process a single image when running on an NVIDIA Tesla K40c GPU.

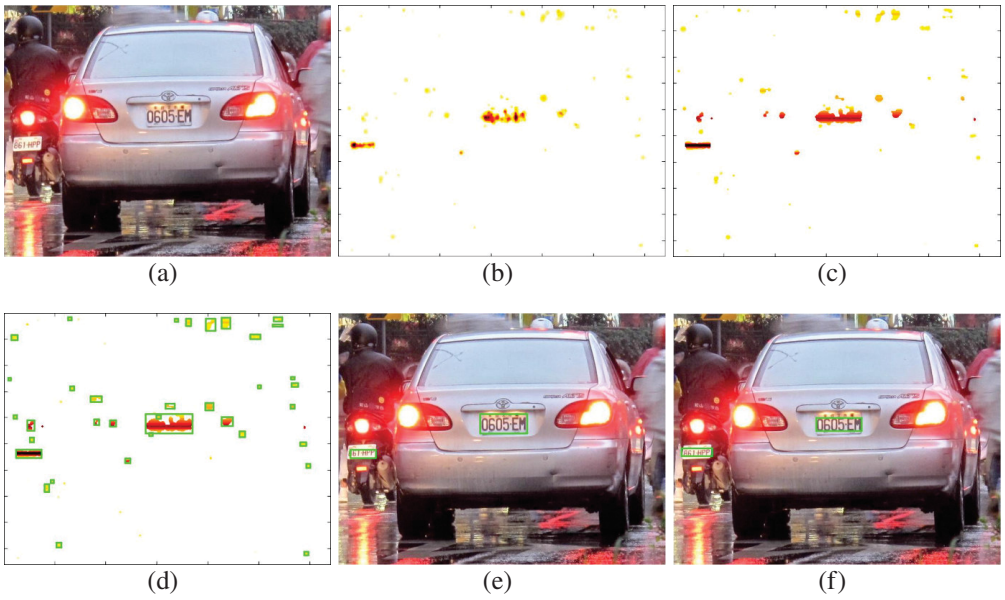


Figure 3.2: The LPD approach proposed by Li et al. (2018). (a) input image (taken from the AOLP dataset); (b) text salience map generated after the sliding window-based detection; (c) text salience map after applying the Non-Maximum Suppression (NMS) and smoothing algorithms; (d) candidate bounding boxes generated by CCA; (e) candidate bounding boxes after the elimination of false positives; and (f) final bounding boxes after box refining and LP/non-LP classification. Image reproduced from (Li et al., 2018).

Liu and Chang (2019) combined handcrafted features with CNNs in their pipeline, which was designed for large visual surveillance scenes and consists of three parts. First, a color-based feature was explored to quickly reject backgrounds with colors other than those of target LPs (in their work, blue LPs from mainland China). Then, for further background rejection, the authors designed a feature that uses information regarding the intensity and color differences between the characters and the background in each LP to express local rectangular features. The AdaBoost algorithm (Viola and Jones, 2004) was employed for both tasks. Lastly, a CNN-based cascade structure containing three distinct networks was proposed to accurately detect the LPs. Their method, which requires 202 ms per image on an NVIDIA GeForce GTX 1060 GPU (i.e., it processes approximately 5 FPS), achieved the highest precision rate and the second-highest recall rate in their assessments with four other LPD methods and two commercial systems. As limitations of their work, we can mention that their method cannot readily be applied to multiple LP layouts, as it leverages color information for background rejection, and that all experiments were performed exclusively on a private dataset.

Mokayed et al. (2021) also explored handcrafted features and CNNs in their pipeline. They combined Discrete Cosine Transform (DCT) and phase congruency to extract a set of candidate LP regions and employed a CNN to eliminate false positives. The authors focused their experimental evaluation on images acquired by drones, which contain several challenges such as large variations in height distance, oblique angles, and many vehicles in a single image (hence, the camera's focus spreads across the vehicles). Although promising results were achieved in images captured by drones, a low F-measure value of 81.1% was obtained in the experiments performed on the Medialab LPR dataset (Anagnostopoulos et al., 2008). As detection rates close to 100% are often reached on Medialab LPR (Bhargav and Deshpande, 2019; Gao et al., 2020a), we conjecture that the thresholds and heuristics of their method were overtuned for drone images, making it not very robust to images acquired by stationary or handheld cameras. The average processing time on an Intel® Core™ i7-8700K CPU was 32 ms per image.

Gonçalves et al. (2018) presented a 15-layer CNN to detect the LPs directly in the input image. The authors showed that even at a high Intersection over Union (IoU) threshold (e.g., 0.7), it is not possible to guarantee that the detected LP encloses all characters (see Figure 3.3). Therefore, they proposed a new loss function that penalizes over-segmented LPs to avoid detections on the inner side of the LP. Their approach was evaluated on public datasets containing Brazilian LPs and worked best on images captured by stationary cameras. According to the authors, this is related to the fact that non-stationary backgrounds contain much more patterns that can be confused with an LP.

Figure 3.3: Three LPs detected with the same IoU value (0.7) with the ground truth; however, only the rightmost has all LP characters completely visible. The ground truth bounding boxes are outlined in blue, while the hypothetical predictions are outlined in orange. Image reproduced from (Gonçalves et al., 2018).

Wang et al. (2022c) reinforced that the speed/accuracy trade-off always accompanies the ALPR's design process and that how to design an effective and efficient ALPR system is still an open-ended question. In this sense, they proposed a compact one-stage LP detector, called VertexNet, with small-resolution input (256×256 pixels) that contains an integration block to extract the spatial features of the LPs as well as a vertex-estimation branch (hence the name of the network) for predicting the geometric shapes of the LPs, which can be later used for LP rectification. Although VertexNet has proven very efficient (i.e., it runs at 5.7 ms per image on an NVIDIA GTX 1080 Ti GPU) and accurate in their experimental evaluation, it probably does not perform well in scenarios where the vehicles are relatively far from the camera, as in the images of the UFPR-ALPR (Laroca et al., 2018) and Vehicle-Rear (Oliveira et al., 2021) datasets, either failing to locate the LPs or predicting many false positives. In fact, we believe this is precisely why the authors forced VertexNet to output only one bounding box per image in their experiments, despite the fact that many real-world applications contain multiple vehicles in the scene (Hsu et al., 2017; Kurpiel et al., 2017; Gonçalves et al., 2018).

Chen et al. (2020) claimed that LPD is easily affected by vehicle detection due to the inclusion relation. Hence, they proposed an end-to-end framework to detect vehicles and LPs simultaneously in a given image, where two separate branches with different convolutional layers were designed for each task. Following (Redmon and Farhadi, 2017; Redmon and Farhadi, 2018), to learn better predictions, the anchor boxes were not selected manually, but using k-means clustering. Finally, attention mechanisms and feature-fusion strategies were employed to improve the detection of small-scale objects. The AP metric and datasets commonly used for general object detection were employed in the experiments. This makes it difficult to compare their method with other LPD approaches in the literature, which generally report the precision and recall rates (considering as correct only the detections with IoU > 0.5 with the ground truth) and perform experiments on datasets created specifically for ALPR-related tasks (Xu et al., 2018; Kessentini et al., 2019; Al-Shemarry and Li, 2020; Lu et al., 2021). Although the authors stated that detecting the vehicles and their LPs in a cascaded fashion is less efficient, their approach presented an inference time of 22 ms on a PC with 4 NVIDIA Titan Xp GPUs, which is longer than the execution times reported in recent cascade-based methods (Silva and Jung, 2020; Laroca et al., 2021b) that also achieved impressive precision/recall rates – this occurs simply because shallower models can be used to detect each LP once the vehicles have been located.

In (Ribeiro et al., 2019; Silvano et al., 2021), the authors highlighted that when a new LP layout is adopted in a country/region, the LPD systems must detect both legacy LPs and those

under the new layout and associated technical specifications. Considering that collecting and manually labeling real-world images of the newly adopted LP layout with sufficient variations can be quite challenging (depending on the transition rules, vehicles with the new LP models remain the exception rather than the rule for a while), the authors presented a methodology for generating synthetic LP images by coupling synthetic images of the target LP layout (in their work, the target was Mercosur LPs) with real-world images containing vehicles with other LP models (e.g., Brazilian), as illustrated in Figure 3.4. The Fast-YOLOv3 model trained exclusively with synthetic images achieved an F-measure of 92% on 1,000 real images from various sources such as search engines, public traffic cameras, and parking lots. The authors considered these results promising; however, it is difficult to assess them accurately since the test images were not made available to the research community.

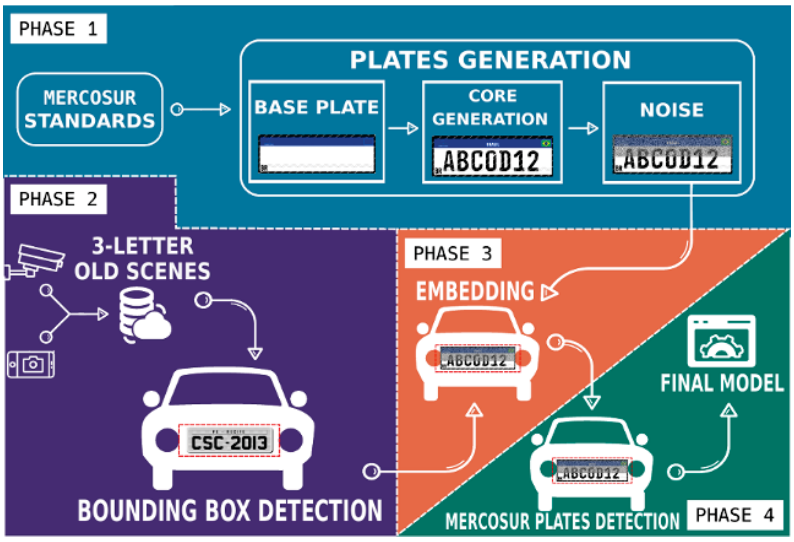


Figure 3.4: Overview of the methodology proposed by Ribeiro et al. (2019) for generating synthetic LP images. Image reproduced from (Ribeiro et al., 2019).

Selmi et al. (2020) slightly modified Mask-RCNN (He et al., 2017) to make it more suitable for LPD. In summary, they removed Mask-RCNN's segmentation module – keeping only the RoIAlign layer – and used a network comparable to GoogLeNet (Szegedy et al., 2015) as the backbone but with fewer inception modules and more pooling layers (in fact, they used convolutional layers with stride = 2). Although promising results were reported in four public datasets, such a network (with an input size of 960×570 pixels) is very computationally expensive, especially considering the real-time requirements of ALPR applications. This limitation was highlighted by the authors themselves and also by Chowdhury et al. (2020).

Chowdhury et al. (2020) stated that most LPD approaches consider the images having a single vehicle in the scene. Thus, they focused on developing a new method for detecting LPs in crowded street scenes, with multiple vehicles at different angles and positions. To enhance the ability to cope with the challenges caused by partial occlusion and varying degree of focus for different vehicles, their method integrates Graph Attention Network (GAT) (Veličković et al., 2018) – using Residual Network (ResNet)-101 (He et al., 2016) for feature extraction – with Progressive Scale Expansion Network (PSENet) (Wang et al., 2019). Their method outperformed both YOLOv2 and PSENet (Wang et al., 2019) in terms of F-measure in three datasets; nevertheless, it takes one second to process a single image on an NVIDIA GeForce GTX 1070 Ti GPU and therefore it cannot be applied to several real-world applications (for comparison purposes, YOLOv2 took only 0.03 seconds in the same setup). The authors also

showed that their method is affected by high/low exposure (i.e., sunlight or shadows on the LPs), failing to differentiate the background and the LPs at the pixel level in these cases.

While also observing that most benchmarks for LPD have only one labeled LP per image, Lee et al. (2022) reinforced that scene texts that look like LPs and arbitrarily shaped LPs are the leading cause of erroneous detections. Hence, they proposed an LP detector that explicitly prevents the learning of non-LP objects (i.e., scene texts but not LPs). As shown in Figure 3.5, their architecture is divided into a backbone network (ResNet-50-FPN) for feature extraction and two parallel sub-networks (i.e., Region Proposal Networks (RPNs)), one for LP detection and other for detecting non-LP objects. The authors added a mutual information term to the objective function for training the networks, expecting the LP detector to maximize the inter-class variation related to non-LP objects. Considering that existing datasets for ALPR do not provide annotations of the scene text (not LP) bounding boxes, the authors introduced a dataset – called LPST-110K – with images/annotations of LPs and non-LP scene-texts to enable the training of their method. The experiments were performed on five public datasets, including LPST-110K. The results showed that their method significantly improved detection performance, especially in terms of precision, which implied that it decreased the number of false positives regarding non-LP scene texts. Nevertheless, it should be noted that the authors unusually reported different metrics for each dataset, making it very difficult to analyze the results. For example, they reported the recall in the UFPR-ALPR dataset, the precision on CCPD, the F-measure in the PKU dataset, and the AP on the LPST-110K. Note that, as detailed in Section 2.1, neither precision nor recall alone can accurately assess the detection quality. Regarding the execution time, considering input images with a resolution of 1280×720 pixels and several LPs per image, their detector runs at 14 FPS on an NVIDIA TITAN X GPU.

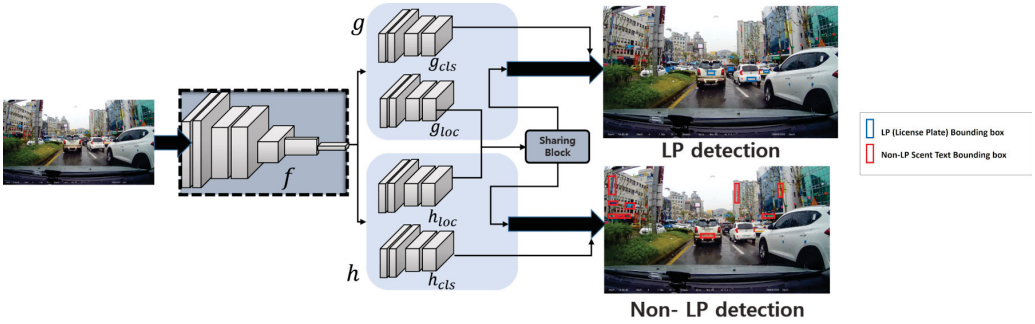


Figure 3.5: Overall architecture of the model proposed by Lee et al. (2022) for LPD. ResNet-50-FPN was employed as the backbone in f. Image reproduced from (Lee et al., 2022).

Aiming to improve the results achieved in the recognition stage, some authors chose to also classify the LPs in some way in addition to detecting them. For example, Laroca et al. (2021b) used a modified Fast-YOLOv2 model to detect the LPs and simultaneously classify their layouts into one of the following classes: American, Brazilian, Chinese, European and Taiwanese[6]. According to their experimental evaluation, carried out on eight public datasets from these five regions, LP layout classification (along with heuristic rules) greatly improved the recognition results since, depending on the LP layout, they avoided errors in characters that are often misclassified and also in the number of predicted characters to be considered. As another example, Xu et al. (2022) proposed a CNN-based detection module that locates the LPs and simultaneously classifies them as having one or two rows of characters. The authors connected this module with another recognition module and reported only end-to-end results. It is worth

[6] Following Laroca et al. (2021b), in this work the "Chinese" layout refers to LPs of vehicles registered in mainland China, while the "Taiwanese" layout refers to LPs of vehicles registered in the Taiwan region.

noting that both detection approaches were designed to be applied to vehicle patches. While Laroca et al. (2021b) applied YOLOv2 (Redmon and Farhadi, 2017) to detect the vehicles in the input images, Xu et al. (2022) took cropped vehicle images as input in their experiments.

Lu et al. (2021) pointed out that most research in LPD is based on individual images, even though there may be multiple frames as input in practical applications. Therefore, they designed an adaptive weight-guided feature aggregation network, called AWFA-LPD, that merges information from adjacent frames to improve LPD results. As shown in Figure 3.6, AWFA-LPD has two branches: one that extracts features from each input frame using ResNet as the backbone and another that obtains optical flow feature maps between adjacent frames using FlowNetSimple (Dosovitskiy et al., 2015). The extracted features are then sent to the aggregation module, which can assign different weights to the feature maps and aggregate them with the feature maps of the reference frame. The weights are assigned based on cosine similarity; the intuition is that feature maps from frames very different from the reference one should have as little impact as possible. Finally, the authors employed R-FCN (Dai et al., 2016) for LPD using the aggregated feature maps. Their method achieved impressive results on the UFPR-ALPR dataset, outperforming five baselines in terms of recall (100%), precision (97.3%) and F-measure (98.6%). As mentioned by the authors, the main shortcoming of their approach is its execution time – to a large extent due to the optical flow module –, which is five times longer than the faster baseline (i.e., 78 *vs.* 16 ms) and three times longer than most of them.

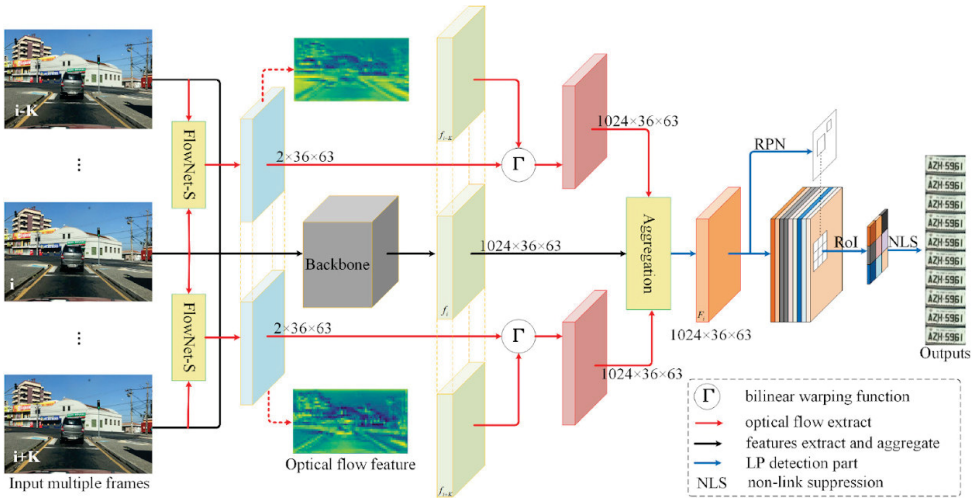


Figure 3.6: The AWFA-LPD framework (Lu et al., 2021). Image reproduced from (Lu et al., 2021).

In the same direction, Zhang et al. (2021a) remarked that existing systems generally focus on single image-based algorithms, yet traffic video sequences provide more practical information than individual frames for ALPR-related tasks. Hence, they proposed a multi-task architecture that integrates LP detection and LP tracking to minimize the additional computational complexity of tracking features generation. This architecture is shown in Figure 3.7. Given an input video, the detector locates the LPs by referring to the temporal relationship between multiple adjacent frames and spatial information in the current frame. At the same time, the tracker generates LP streams and assigns them different identities using motion information and discriminative features. The EAST scene text detector (Zhou et al., 2017) was used as the detection backbone. In an experimental evaluation conducted on three public datasets with Brazilian LPs, their method reached better precision and recall rates than several baselines that process frames individually. Although computation complexity was reduced by sharing feature extraction and avoiding repeated calculations in a separated tracking stage, their method's main limitation is its

computational complexity as it takes about 122 ms to process each frame; for comparison, one of the baselines achieved a slightly lower F-measure (e.g., 98.5% vs. 99.3% on the UFPR-ALPR dataset) taking only a tenth of that time to process each frame on similar hardware.

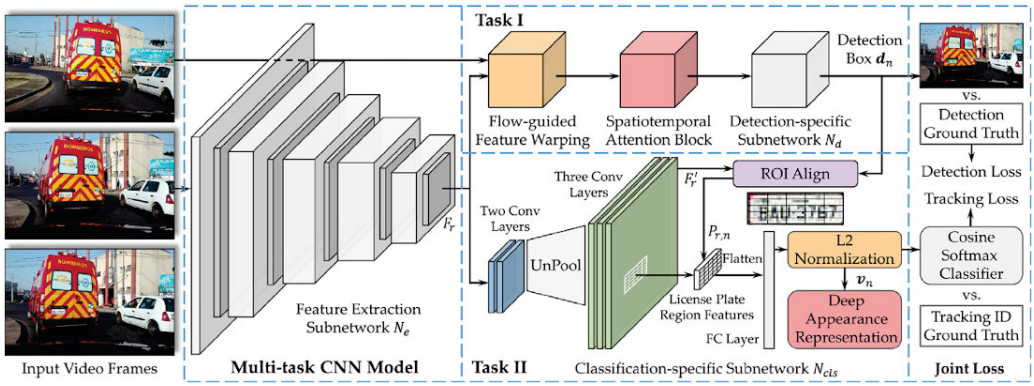


Figure 3.7: The multi-task architecture proposed by Zhang et al. (2021a) that integrates LP detection and LP tracking. Image reproduced from (Zhang et al., 2021a).

3.2 License Plate Recognition (LPR)

The great speed/accuracy trade-off provided by YOLO networks (Redmon et al., 2016; Redmon and Farhadi, 2017; Redmon and Farhadi, 2018; Bochkovskiy et al., 2020; Wang et al., 2021a) inspired many authors to explore similar architectures targeting real-time performance for LPR. For example, Silva and Jung (2017) proposed a YOLO-based model that simultaneously detects and recognizes all characters within a cropped LP (we depict how object detectors handle OCR tasks in Figure 3.8). This model, called CR-NET, consists of the first eleven layers of YOLO and four other convolutional layers added to improve nonlinearity. While impressive FPS rates – i.e., 448 FPS on an NVIDIA Titan X GPU – were attained in experiments carried out in the SSIG-SegPlate dataset (Gonçalves et al., 2016a), less than 65% of the LPs in the test set were correctly recognized. According to the authors, the bottleneck of their approach was in letter recognition since the character classes (in particular, letters) are highly unbalanced in the training set of the SSIG-SegPlate dataset (as in most datasets for ALPR (Zhang et al., 2021c)).

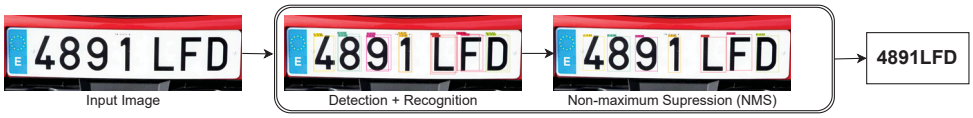


Figure 3.8: An illustration of how object detectors (e.g., CR-NET) handle OCR tasks. First, the characters are simultaneously detected and recognized. Then, an NMS algorithm eliminates redundant detections (e.g., those with IoU ≥ 0.25) since the network often detects the same character more than once. Finally, the detections are sorted based on some predefined criteria (e.g., x-coordinate for single-row LPs) to produce the final string.

Taking this into account, Silva and Jung (2018) generalized CR-NET by retraining it with an enlarged training set composed of real and artificially generated images using font-types similar to the LPs of the target regions (i.e., Brazil, Europe, and the United States), as shown in Figure 3.9. The retrained network became much more robust for detecting and classifying real characters on Brazilian LPs and also on LPs from other regions, outperforming previous works and commercial systems in three public datasets. In (Silva and Jung, 2020), in a very similar way, the same authors retrained the CR-NET model with a massive number of artificial images generated by blending real LPs with synthetic characters through Poisson blending (Pérez et al.,

2003). Impressive results on several public datasets have been achieved through CR-NET in recent works (Laroca et al., 2021b; Oliveira et al., 2021; Silva and Jung, 2022).

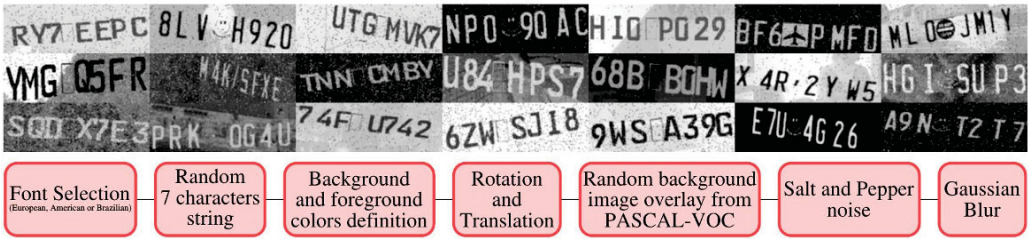


Figure 3.9: Artificial LP samples generated by Silva and Jung (2018). Such LPs use font-types similar to the LPs of the target regions (i.e., Brazil, Europe, and the United States), which made their network more robust for detecting and classifying real characters of LPs issued in those regions. Image reproduced from (Silva and Jung, 2018).

While some authors (Kessentini et al., 2019; Lee et al., 2019; Kim et al., 2021) employed YOLO models without any change or refinement for LPR, Henry et al. (2020) applied YOLOv3-SPP – a version of YOLOv3 (Redmon and Farhadi, 2018) with Spatial Pyramid Pooling (SPP) – to this task. They developed an algorithm to determine whether the detected characters are arranged in one or two rows, regardless of the LP layout. Although their approach achieved high recognition rates on five datasets from multiple countries/regions, the YOLOv3-SPP model is excessively deep for LPR (i.e., it has more than 100 layers), making it difficult for the whole system to meet the real-time requirements of ALPR applications – especially if there are multiple vehicles in the scene –, as each LP is recognized individually.

Instead of exploring object detectors, Li et al. (2018) handled LPR as a sequence labeling problem, i.e., without character-level segmentation. First, sequential features were extracted from the entire LP patch using a 9-layer CNN in a sliding window manner. Then, Bidirectional Recurrent Neural Networks (BRNNs) with Long Short-Term Memory (LSTM) were applied to label the sequential features. Lastly, Connectionist Temporal Classification (CTC) was employed for sequence decoding. Figure 3.10 illustrates the overall structure of their approach, which attained better recognition rates than the two baselines chosen by the authors. Nevertheless, only Taiwanese LPs were used in the experiments, and the execution time was not reported.

Wang et al. (2018a) rectified the LP images prior to the recognition stage so that all LPs have a uniform orientation and thus are easier to recognize. They employed a Spatial Transformer Network (STN) (Jaderberg et al., 2015) for this task. Then, in a very similar way to the approach presented by Li et al. (2018), they extracted sequential features using a CNN model (based on VGG (Simonyan and Zisserman, 2015)), adopted a BRNN to output labels from the sequential features, and applied CTC to decode the sequential labels and produce the final recognition results. Their method (see Figure 3.11), pre-trained on synthetic LPs (created using OpenCV) and fine-tuned on real Chinese LPs, achieved better results compared to the baseline (Li et al., 2019) and took approximately 17.5 ms to recognize an LP on an NVIDIA 1080 Ti GPU. No public datasets were used in their experiments.

Zou et al. (2020) also adopted a Bi-directional Long Short-Term Memory (Bi-LSTM) network (Graves and Schmidhuber, 2005b,a) to implicitly locate the characters on each LP. They explored a 1-D attention module to extract useful features of the character regions, improving the LPR performance. Their experiments were performed on four public datasets: AOLP (Hsu et al., 2013), PKU (Yuan et al., 2017), CCPD (Xu et al., 2018) and CLPD (Zhang et al., 2021c). Their network achieved better results than the baselines on the three datasets with LPs from mainland China; however, the comparison of their method with others in the AOLP dataset should not be

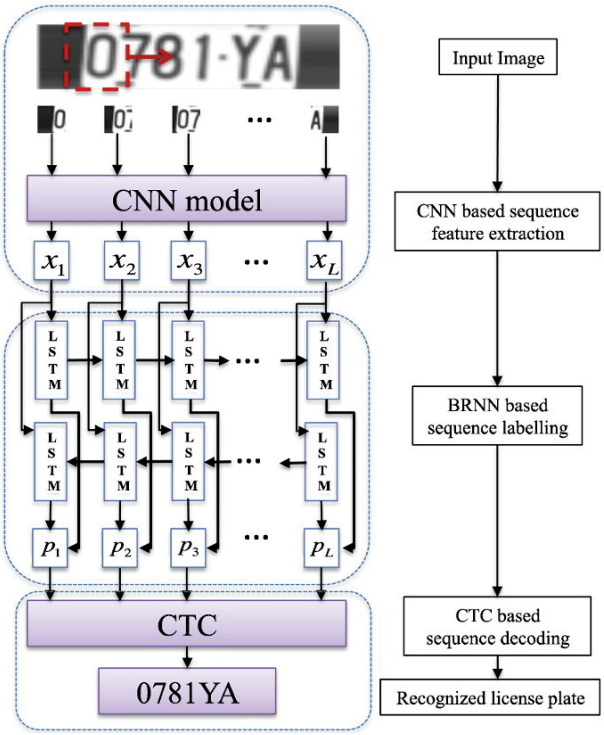


Figure 3.10: The sequence labeling-based approach proposed by Li et al. (2018) for LPR. First, a 9-layer CNN extracts sequential features in a sliding window manner. Then, BRNNs with LSTM are used for sequence labeling. Lastly, CTC is employed for sequence decoding. Image reproduced from (Li et al., 2018).

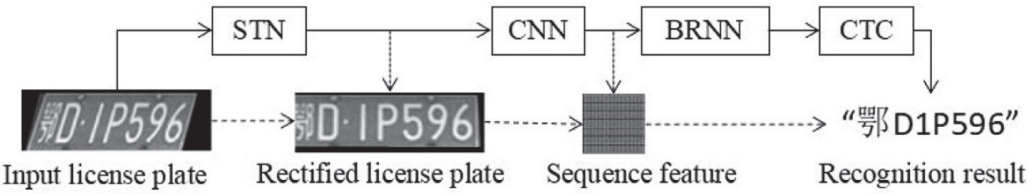


Figure 3.11: The LPR approach proposed by Wang et al. (2018a), which is capable of recognizing English letters, digits, and Chinese characters. Image reproduced from (Wang et al., 2018a).

considered as the authors adopted a different evaluation protocol from that used by the baselines. Details were not provided regarding the execution time of their approach.

Similarly, Zhang et al. (2021c) used a 2-D attention mechanism to optimize their OCR model, which uses a 30-layer CNN based on Xception (Chollet, 2017) for feature extraction. An LSTM model was adopted to decode the extracted features into LP characters. The authors highlighted that it is difficult to manually collect LP images from various regions, which makes most ALPR datasets heavily biased toward specific regional identifiers. Therefore, they explored the asymmetric CycleGAN model – proposed in their previous work (Zhang et al., 2019b) (see Section 3.3) – to synthesize images of Chinese LPs with different transformations and balanced character classes, reducing data bias and improving model generalization ability. The proposed method outperformed all baselines in four public datasets – AOLP (Hsu et al., 2013), PKU (Yuan et al., 2017), CCPD (Xu et al., 2018) and CLPD (Zhang et al., 2021c) – especially with limited training data. Although the authors claimed that their approach does not leverage any heuristic rules or post-processing, they trained a recognition network specifically for each LP layout unlike some recent works (e.g., (Laroca et al., 2021b; Silva and Jung, 2022)), which employed a single model for LPs from different regions. In other words, their network implicitly learns heuristic rules about each LP layout. For example, when trained using images from the CCPD dataset

(which contains LPs from mainland China), it learns to always predict a Chinese character as the first LP character since this is the case in every single training example. The authors did not report information about the execution time of their approach.

Several works also designed multi-task CNNs to process the entire LP image holistically, circumventing character segmentation. For instance, Špaňhel et al. (2017) focused on recognizing LPs in low-resolution and low-quality images, where segmentation becomes challenging due to blurred characters. Their model initially processes the entire image using convolutional layers. Then, eight separate branches with fully connected layers predict up to eight characters (including a "non-character" class) for specific positions on the LP (see Figure 3.12). Their model, often referred to as Holistic-CNN (Meng et al., 2018; Gong et al., 2022; Liu et al., 2024a), achieved a processing speed of over 1000 FPS on an NVIDIA GeForce GTX 1080 GPU and outperformed two commercial systems – namely, OpenALPR (OpenALPR, 2024) and UnicamLPR (CAMEA, 2024) – on three public datasets containing Czech LPs.

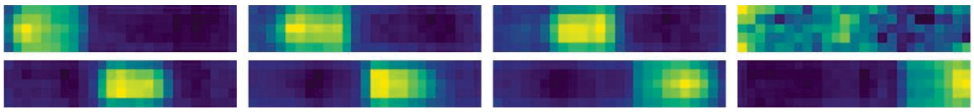


Figure 3.12: The attention of Holistic-CNN's fully connected layers for different characters on a Czech LP. From left to right, top to bottom: 1st to 8th character. In most cases (i.e., on Czech LPs with less than eight characters), the 4th position does not contain any character (it is blank). Image reproduced from (Špaňhel et al., 2017).

A similar approach was introduced by Gonçalves et al. (2018), who designed a multi-task CNN with 14 layers to locate and recognize all LP characters simultaneously. Promising results (in terms of both accuracy and execution time) were achieved in two public datasets with Brazilian LPs by massively taking advantage of synthetic data. The same authors explored a very similar multi-task model in (Gonçalves et al., 2019). However, they focused on designing a novel strategy to generate synthetic data and thus improve the LPR results obtained by the multi-task model in low-resolution LP images (we describe this latter work in Section 3.3).

Wang et al. (2022c) observed that these multi-task models for LPR employ fully connected layers as classifiers to recognize the characters on the predefined positions of the LPs. Hence, without making massive use of synthetic data, they may not generalize well with small-scale training sets since the probability of a specific character appearing in a specific position is low; in fact, a given character may never appear in a specific position on a small set of LPs. Thus, they proposed a weight-sharing classifier for LPR, called SCR-Net, which can spot instances of each character across all positions. Figure 3.13 shows three weight-sharing classifiers used by the authors for the three types of characters on the LPs from the CCPD dataset (Xu et al., 2018) (Chinese characters, English letters, and digits). The authors explored an encoding technique to vertically squeeze feature maps into 1-D horizontal features (32×1), before feeding them to the classifier. Despite running relatively fast, taking 5.7 ms to process each image on an NVIDIA GTX 1080 Ti GPU, their approach reached better results than all baselines on four public datasets: AOLP (Hsu et al., 2013), PKU (Yuan et al., 2017), CCPD (Xu et al., 2018) and CLPD (Zhang et al., 2021c). One of their method's limitations is that a new training process must be carried out for each LP layout to be recognized. For example, the authors trained and tested two instances of their model in the experiments: one for LPs from mainland China and one for LPs from the Taiwan region. Moreover, we conjecture that their approach is not as robust – or even does not work – for LPs with two rows of characters due to the left-to-right horizontal encoding technique employed.

Zhang et al. (2021d) also pointed out that existing multi-task models – including those proposed by Špaňhel et al. (2017) and Gonçalves et al. (2018) – cannot exploit the diversity

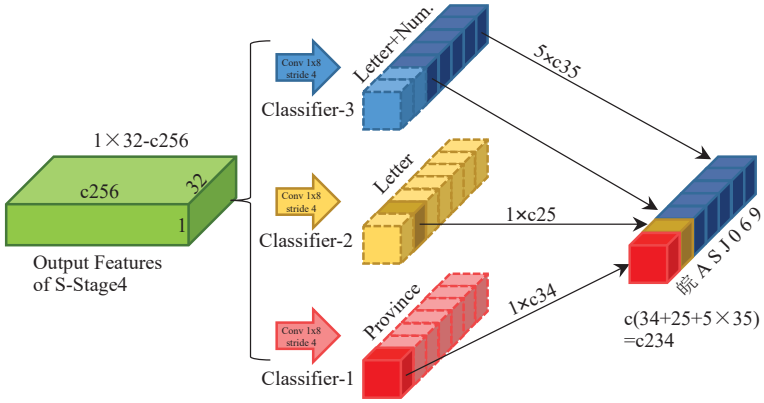


Figure 3.13: Example of weight-sharing classifiers for Chinese LPs. Image reproduced from (Wang et al., 2022c).

of LP characters at different positions. Thus, in the same way as Wang et al. (2022c), they employed a shared classifier to recognize the characters at different positions in a unified way. For producing more discriminative features, the authors explicitly disentangled the semantic and position information of the LP characters using two networks in parallel, with supervision on each of them being optional (see Figure 3.14). For the semantic network, the ground truth corresponds to a bounding box for each character with the pixels annotated (i.e., colored) according to the semantic class of that character. Similarly, the ground truth labels for the position network are also represented with bounding boxes; however, the pixels in each bounding box are determined by the position of the respective characters in the LP. The semantic and position networks connect the same backbone network – BiSeNet (Yu et al., 2018) – to share global features and produce the semantic and position features by appending different heads. Based on experiments performed on four datasets (Medialab LPR, AOLP, CLPD, and CCPD), the authors noted that more supervision signals are useful as promising results were achieved in all of them. Nevertheless, it is unclear whether their method generalizes well to unseen data as they trained an instance of their network specifically for each dataset or LP layout. The same is true for LPs with two rows of characters, as all experiments were performed on single-row LPs. It is worth noting that the authors discarded 175 images from the AOLP dataset in their experiments; therefore, the results reported on it are not comparable with those obtained in previous works (which did not discard any image). Finally, regarding execution time, different models were explored as the base network in the backbone (i.e., ResNet-18, -34, -50, and -101), thus enabling different speed/accuracy trade-offs. In this way, depending on the model chosen as the backbone, their network processes between 57 and 191 FPS in the AOLP dataset on an NVIDIA GTX 1080 Ti GPU.

Zeni and Jung (2020) highlighted that detection-based recognition methods (e.g., CR-NET) tend to adapt better to different LP layouts since they learn each character's appearance separately, while segmentation-free approaches (e.g., Holistic-CNN) alleviate the cost of manually labeling the bounding box of each character on the LP. Thus, they presented a Weakly Supervised Character Detection (WSCD) approach that explores the best of both worlds: it uses only string-level annotations to learn the characters' bounding boxes in a weakly supervised fashion. Their approach is built on top of the multiple instance detection network proposed by Bilen and Vedaldi (2016), with an instance-aware online refinement approach, a knowledge distillation module, and a sub-network for estimating the number of characters to guide the final recognition result. Although their method produced impressive results for some datasets, the module that classifies the number of characters showed signs of overfitting (according to the authors themselves); thus, very low recognition rates – compared to baselines – were obtained in some other datasets.

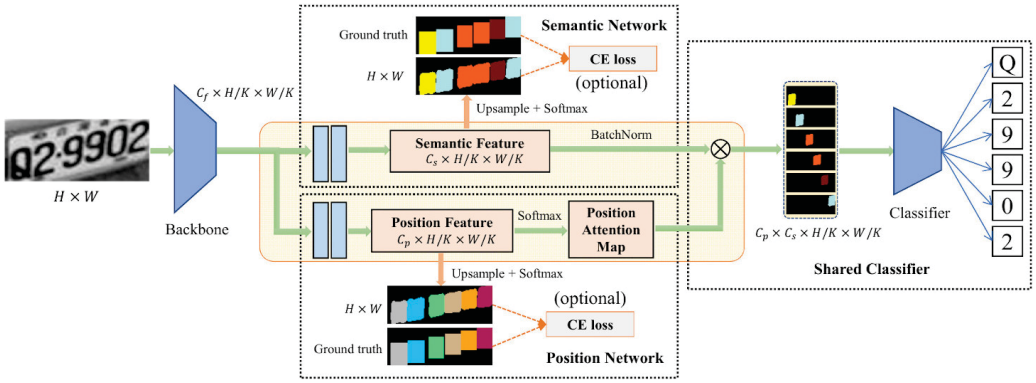


Figure 3.14: Illustration of the LPR method proposed by Zhang et al. (2021d). It comprises four main components: a backbone network, a semantic network, a position network, and a shared classifier. C_p and C_s are the number of characters in an LP and the number of character classes, respectively. Image reproduced from (Zhang et al., 2021d).

Zhang et al. (2020a) reinforced that robust and efficient LPR is still an urgent task to be solved. In addition, they stated that there are only a few video-based approaches modeling temporal information explicitly (as illustrated in Figure 3.15). Accordingly, they proposed a quality-aware algorithm that first evaluates the image quality of each LP patch and then recommends the recognition result predicted in the highest quality frame as the final decision. The authors employed knowledge distillation (Hinton et al., 2014) to compress their quality awareness network and make it lightweight. Although impressive recognition results were reported in the UFPR-ALPR dataset (Laroca et al., 2018), they are not directly comparable with those reached in other works since the authors expanded/modified the original test set through data augmentation (instead of just augmenting the training set); the authors also carried out experiments with Chinese LPs, but as they belong to a private dataset it is difficult to assess the reported results. Furthermore, even though the authors emphasized the efficiency requirements of LPR, their approach cannot process 30 FPS (even with the LPD stage not being addressed), and details about the hardware used in their experiments were not provided. In subsequent work (Zhang et al., 2021a), the same authors integrated this quality-aware algorithm (with a few changes; for example, without knowledge distillation) into an end-to-end framework, thus reaching better results in terms of recognition rate than several baselines that process frames individually in three video-based public datasets: SSIG-SegPlate (Gonçalves et al., 2016a), LQPV (Seibel et al., 2017) and UFPR-ALPR (Laroca et al., 2018). While such a quality-aware approach is very appealing for multi-frame LPR in conventional ALPR applications (e.g., traffic law enforcement), it is not able to handle the challenging cases – yet common in forensic applications – where a vehicle's LP is illegible or has very low quality in every frame of a video because it was recorded by cameras installed for purposes other than ALPR.

Vašek et al. (2018) extended the CNN model proposed in (Goodfellow et al., 2014a), originally designed for number recognition on street view images, to process a sequence of rectified LP images obtained from a tracker and output a distribution over a set of LP strings. They addressed a relatively under-explored scenario of when the input of the LPR system is a low-resolution video captured by an ordinary camera or a cell phone. As illustrated in Figure 3.16, their architecture has three components: (i) a CNN that extracts features from each image in the sequence; (ii) an aggregation layer that shrinks the feature sequence into a distribution over strings; and (iii) another CNN that converts the output of the aggregation layer into a distribution over strings. It is noteworthy that the number of images in the test sequences can be arbitrary thanks to the aggregation layer. Empirical evaluation on low-resolution European LPs (mostly Czech) showed that their approach significantly outperformed both baseline methods and human

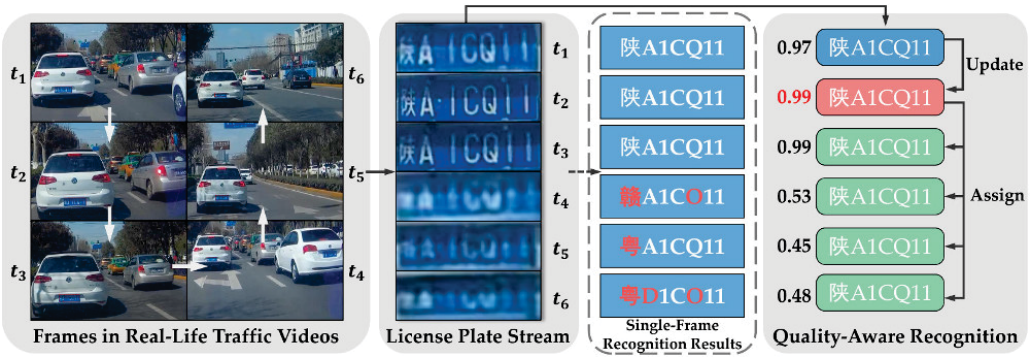


Figure 3.15: Many frames are involved with the same LP at different times in a traffic video. The red characters of the single-frame recognition results indicate incorrect predictions that can be avoided through the quality-aware approach proposed by Zhang et al. (2020a). Image reproduced from (Zhang et al., 2020a).

performance. Nevertheless, the experiments were performed on a proprietary dataset only, with 8.3 million image sequences (each having five images) being used for training their networks. No experiments related to execution time were reported.

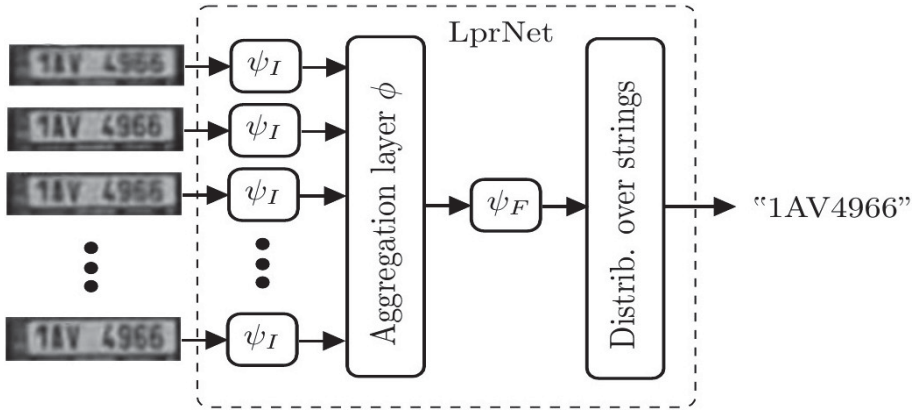


Figure 3.16: The LPR approach proposed by Vašek et al. (2018), which takes a sequence of rectified LP images as input. Image adapted from (Vašek et al., 2018).

Zhuang et al. (2018) proposed a semantic segmentation technique followed by a character count refinement module to recognize the characters of an LP. Figure 3.17 illustrates their framework. For semantic segmentation, they simplified the DeepLabV2 (ResNet-101) model (Chen et al., 2018) by removing the multi-scaling process, thus increasing computational efficiency. According to the authors, the purpose of the multi-scaling process is to fuse hierarchical global information; however, the semantic areas of different characters have a lower correlation in the LPR task. After obtaining the LP semantic map, the character areas were generated through CCA. Finally, Inception-v3 (Szegedy et al., 2016) and AlexNet (Krizhevsky et al., 2012) were adopted as the character classification and character counting models, respectively. The authors claimed that both an outstanding recognition performance and a high computational efficiency were attained. Nevertheless, they assumed that LPD is easily accomplished and used cropped patches (from the ground truth) containing only the LP with almost no background as input. In addition, their approach cannot process images in real time (it processes 25 FPS on an NVIDIA TITAN X GPU), especially when considering the time required for the LPD stage, which is generally more time-consuming than the recognition one. Lastly, they trained specific models for each LP layout (i.e., the experiments on Greek and Taiwanese LPs were conducted separately); therefore, adding support for a new layout requires retraining the networks.

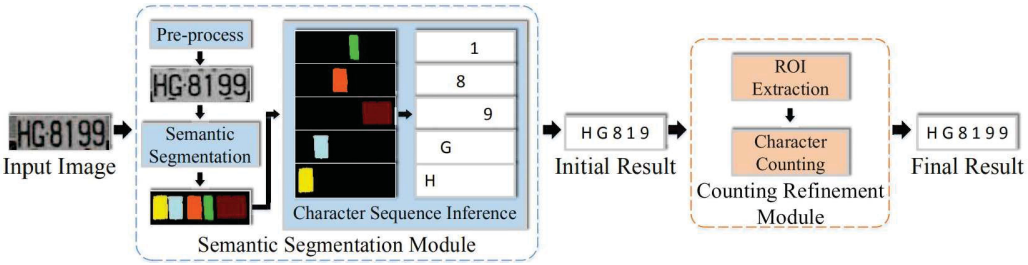


Figure 3.17: Illustration of the framework proposed by Zhuang et al. (2018) for LPR. Their framework consists of two key modules: semantic segmentation and counting refinement. The former produces the semantic map and the initial character sequence, while the latter generates the final result (i.e., the LP text) through counting characters. Image reproduced from (Zhuang et al., 2018).

Selmi et al. (2020) used Mask-RCNN (He et al., 2017) for LPR. The network was trained to predict 37 classes (0-9, A-Z, and one Arabic word). Some post-processing rules were applied to the network's output to improve the recognition results (e.g., predicted regions too wide or too small to be a character were discarded). Despite the fact that promising results were reported in four public datasets, the chosen model (with an input size of 530×300 pixels) is much more computationally expensive than those used in other works – e.g., (Silva and Jung, 2020; Liu et al., 2021; Zhang et al., 2021d) – for this task, which makes it difficult (or even impossible) for it to be employed in some real-world applications (especially those where multiple vehicles can coexist on the scene). The authors themselves highlighted this limitation in their method.

Liu et al. (2021) observed that most recognition methods were proposed for single-row LPs, considering LPR a one-dimensional sequence recognition problem. They stated that these methods are not suitable for recognizing two-row LPs because the features of adjacent characters may get mixed up when directly transforming an LP image into a one-dimensional feature sequence. In an attempt to solve this problem, they proposed a 2-D spatial attention module to recognize LPs from a two-dimensional perspective (see Figure 3.18). The authors adopted the backbone from Holistic-CNN (Špaňhel et al., 2017), with a few modifications, to extract visual features from the input image. Unlike Zhang et al. (2021c), who also explored a 2-D attention module, Liu et al. (2021) adopted one fully connected layer (i.e., a shared classifier) as the decoder and not a recurrent structure. Their method performed better than several baselines on images from three private and two public datasets containing Chinese LPs. While much of the authors' focus was on recognizing two-row LPs, they overlooked public datasets containing images of LPs with two rows of characters – some examples are the EnglishLP, UFPR-ALPR and Vehicle-Rear datasets – and evaluated their network exclusively on two-row LPs from private datasets. Their network can process 278 FPS on an NVIDIA GTX 1080 Ti GPU.

In (Xu et al., 2022), an extension of (Xu et al., 2021), the authors also emphasized that most methods for ALPR can only handle single-row LPs. In this way, similar to (Zhang et al., 2021c; Liu et al., 2021) and inspired in (Wojna et al., 2017), they adopted a 2-D attention mechanism for LPR where the encoder is a lightweight CNN structure and the decoder is attention-based. A Gated Recurrent Unit (GRU) (Cho et al., 2014) was used to convert the feature maps into a character sequence. Before recognition, each detected LP is fed into a feature alignment module based on perspective transformation prediction and grid sampling, which rectifies the deformed LP features into regular ones. Their method was primarily evaluated on Chinese LPs, considerably outperforming other well-known models for recognizing LPs from mainland China. However, we remark that the authors fine-tuned their method (and not the baselines) on 150k images from a private dataset. The downside of their method lies in its efficiency since, according to the authors, it is about two times slower than CTC-based models

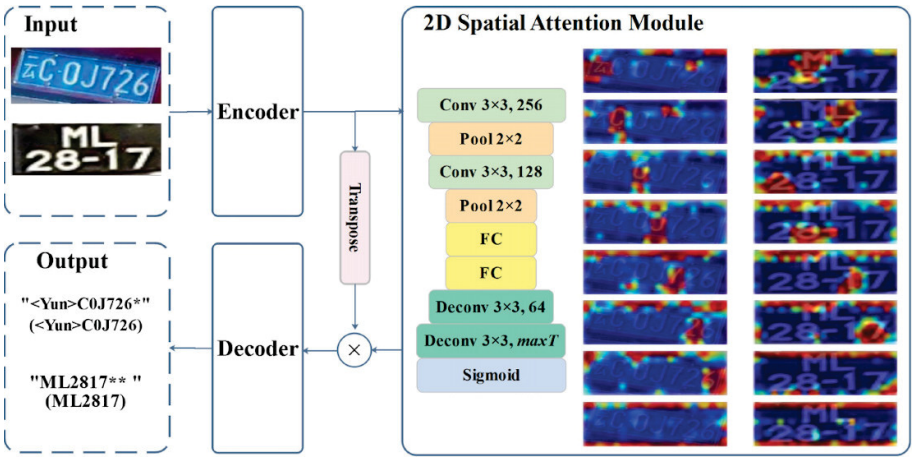


Figure 3.18: The overall architecture of the network proposed by Liu et al. (2021) for LPR. "*maxT*" is the max length of LP texts in the training set (in their experiments, $maxT = 8$), and "$*$" in the results represents a blank character (to handle LPs with different numbers of characters). Image reproduced from (Liu et al., 2021).

– which are already known in the ALPR literature for being quite time-consuming (Zhang et al., 2021d; Liu et al., 2021).

Lee et al. (2022) explored a similar method for LPR, in which the encoder network (with seven convolutional layers) is followed by Bi-LSTM, and an attention mechanism with GRU and LSTM is employed as the decoder. Although the authors performed experiments on five public datasets from four different regions, they described this approach and also reported its results very superficially, as the focus of their work was on the LPD stage.

3.3 Synthetic Data

As highlighted in Section 2.3, it is well-known that unbalanced data is undesirable for neural network classifiers since the learning of some patterns might be biased. This problem is even more pronounced in the ALPR context, particularly in the LPR stage, as it is difficult to manually collect LP images from a variety of regions, which makes most existing ALPR datasets heavily biased toward specific regional identifiers (Zhang et al., 2021c; Liu et al., 2021).

Considering the above discussion, many data augmentation methods have been proposed in the ALPR context to eliminate bias from the experiments and reduce the number of real images needed for training deep models (Gonçalves et al., 2018; Silva and Jung, 2020; Laroca et al., 2021b). To narrow the scope of this section, we focus on describing relevant works where the authors exploited generative models (mostly GANs) to this end.

Although GANs were proposed in 2014 (Goodfellow et al., 2014b), it was not until 2017 that they were first applied to data augmentation in the ALPR context. Wang et al. (2017) pointed out that LP images are hard to collect due to privacy issues and regional characteristics (i.e., the LPs differ in countries and regions). Therefore, they trained CycleGAN (Zhu et al., 2017b) with the Wasserstein distance loss (Arjovsky et al., 2017) to learn a mapping that maps script images (Figure 3.19a) into real images (Figure 3.19b). They used the generated images (Figure 3.19c), which are labeled, to pre-train a Convolutional Recurrent Neural Network (CRNN) model (Shi et al., 2017) for recognizing Chinese LPs. The CRNN model was then fine-tuned on real images. The authors reported many experiments, which demonstrated that this strategy (i.e., pre-training an OCR model on synthetic data created by CycleGAN and fine-tuning it on real data) brings significant improvements in terms of recognition rate. For example, the CRNN model pre-trained on CycleGAN images and fine-tuned on 9,000 real images reached better results than the same

model trained on a set containing 50,000 real images without CycleGAN-generated images. The major shortcoming of their work is that only private datasets (with tens or hundreds of thousands of training images) were used in the experiments. In addition, the authors trained one model to generate blue LPs and another to generate yellow LPs, without detailing why not train a single model to generate LPs of both colors (we conjecture that the LP images generated in this way have artifacts). Lastly, it is important to note that this work is only available on arXiv[7]; that is, it has not gone through the peer-review process. Still, we chose to describe it here since it is the first work applying GANs to generate LP images and because some of the authors have already published relevant articles in the ALPR context (Li et al., 2018, 2019; Zhang et al., 2021c).

(a) Script images (b) Real images (c) CycleGAN images

Figure 3.19: Wang et al. (2017) trained CycleGAN (Zhu et al., 2017b) to generate images of Chinese LPs (c). The CycleGAN model was trained using images created by a script (a) (i.e., colors and character deformations were hard-coded) as one domain and real images (b) as another. Image reproduced from (Wang et al., 2017).

Exactly the same strategy was adopted shortly after by Zhang et al. (2018b). That is, they also employed CycleGAN (Zhu et al., 2017b) (with the original loss) to automatically generate a large number of Chinese LPs for pre-training the CRNN model (Shi et al., 2017). As in (Wang et al., 2017), the CycleGAN-generated images did help improve the performance of the OCR model (from 95.5% to 97.6%), and the experiments were conducted exclusively on a private dataset. The authors classified the need for an abundant source of training images as the main limitation of this approach since they tried to generate images of American LPs using the Caltech Cars dataset (Weber, 1999) (which has 126 images), but the results were not satisfactory.

Wu et al. (2018) emphasized that there is no clear common understanding of how many labeled LPs are needed to train a recognition model that achieves satisfactory performance. They tried to address such a question by analyzing the performance of a recognition model based on DenseNet (Huang et al., 2017) when trained on a few real images and many artificial ones. In the same direction as (Wang et al., 2017; Zhang et al., 2018b), they explored CycleGAN (Zhu et al., 2017b) – with gradient penalty (Gulrajani et al., 2017) – to learn the mapping relationship between script LPs and real LPs. However, differently from what was done in those works, the authors used both generated and real images to train the recognition model from scratch (rather than pre-training it on generated images and then fine-tuning it on real images). The results showed that their recognition model trained from scratch on only 300 real images in addition to hundreds of thousands of generated images reached competitive results to the CRNN model pre-trained on generated images and fine-tuned on 200,000 real images by Wang et al. (2017). Although these results are quite promising, the experiments were performed exclusively on images from a private dataset. We conjecture that the test set is not challenging enough, with many "easy" LP images and a few difficult ones. This would explain the high recognition rates being achieved with only 300 real training images and not improving when the number of real LPs is increased from 4,750. It is worth noting that the authors performed experiments on images from the AOLP dataset (Hsu et al., 2013), but they did not generate Taiwanese LPs for training their recognition model (they explored only real images with simple data augmentation techniques such as affine transformation,

[7] arXiv (`https://arxiv.org/`) is an open-access repository of electronic preprints, with a submission rate of over 19,000 articles per month as of February 2024 (arXiv, 2024).

erosion, and dilation). This reinforces what was concluded by Zhang et al. (2018b), i.e., that such generative models need a large training set to produce satisfactory results.

Zhang et al. (2019b) remarked that it is difficult to manually collect images of LPs from different states/provinces across a country. To better illustrate, they noted that 95% of the images from the CCPD dataset (Xu et al., 2018) were captured in a single city in China, so the first two characters on different LPs are usually the same[8]. In this way, according to the authors, a recognition model trained on CCPD's images – without some kind of data generation – cannot be used nationwide. Their approach has two main differences from those already described in this section. First, the authors trained the CycleGAN model (Zhu et al., 2017b) without the second cycle-consistency loss (i.e., they discarded the loss responsible for mapping real images into synthetic ones) – this is why this model ssers termed as asymmetric CycleGAN in a subsequent work (Zhang et al., 2021c). Second, they trained multiple networks to generate images with specific characteristics. For example, they trained one CycleGAN network specifically to map script images (Figure 3.20a) into bright LPs (Figure 3.20b), another to map script images into dark LPs (Figure 3.20c), and so on. The generated images consistently improved the results obtained on the CCPD dataset by a CNN based on Xception (Chollet, 2017), even though the authors acknowledged that CycleGAN does not handle character details very well. The experimental evaluation could have been more extensive since the authors did not detail how discarding CycleGAN's second cycle-consistency loss affected the quality of the generated images, nor whether it would be possible to train a single CycleGAN-based network to generate LP images with different characteristics.

(a) Script LPs

(b) Bright LPs

(c) Dark LPs

Figure 3.20: Zhang et al. (2019b) trained multiple CycleGAN-based networks (Zhu et al., 2017b) to generate LP images with different characteristics (b) (c). Each network was trained using script images (a) as one domain and real images with specific characteristics as another domain. Image reproduced from (Zhang et al., 2019b).

Wu et al. (2019) argued that existing models at that time could transfer general color and texture from the source images to target images but ignored the structural properties of each character region, yielding blurry and distorted results. Therefore, they proposed PixTextGAN, which comprises a generator, a discriminator, and a text recognition module to generate realistic LP images while preserving character structure information. As illustrated in Figure 3.21, considering paired data, PixTextGAN is trained using a structure-sensitive loss function that integrates pixel-wise loss (i.e., Mean Squared Error (MSE)), content loss (similar to perceptual loss, but the feature representations are extracted by a pre-trained text recognition network), and CTC loss (Graves et al., 2006). The authors compared PixTextGAN with CycleGAN (Zhu et al., 2017a) and pix2pix (Isola et al., 2017) in the ReId (Špaňhel et al., 2017) and CCPD datasets (Xu et al., 2018). To this end, they pre-trained CRNN (Shi et al., 2017) on 100,000 generated images

[8] The first character in Chinese LPs denotes the province to which the vehicle is registered, while the second character is a letter indicating the issuing city within that province (Xu et al., 2018, 2021; Zhang et al., 2019c, 2021c).

and fine-tuned it on different numbers of real images. According to the qualitative and quantitative results, PixTextGAN outperformed CycleGAN and pix2pix in both datasets. As PixTextGAN exploits a recognition module to improve the generation of LP images, it remains to be analyzed whether a single model would be able to generate LPs from different regions and with different characteristics (they trained two distinct models, one for Chinese LPs and another for Czech LPs).

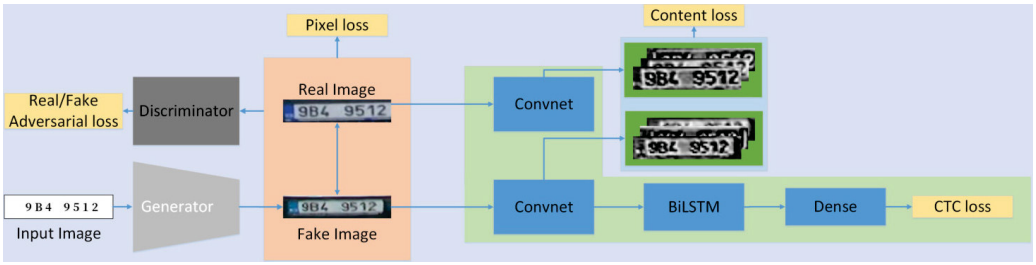


Figure 3.21: The framework of the PixTextGAN model (Wu et al., 2019), which aims to generate realistic LP images while preserving text order consistency between synthetic and real images. Image reproduced from (Wu et al., 2019).

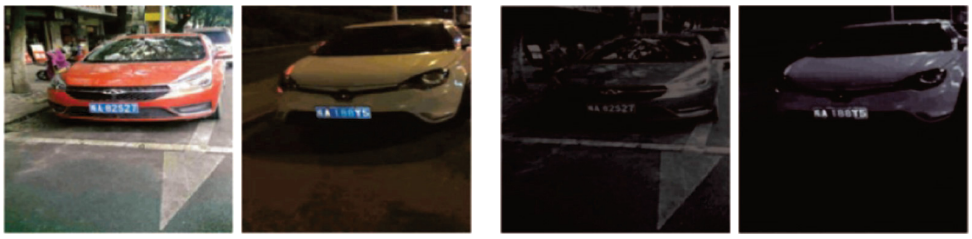
Han et al. (2020) listed several public datasets for ALPR (the best-known ones), observing that none contain images of Korean LPs. To train a recognition model for these LPs, they tried to build a large-scale dataset through web-scraping but managed to find only 159 images of Korean LPs. Considering this, they proposed using image-to-image translation GANs to generate images of Korean LPs from script images. They trained CycleGAN (Zhu et al., 2017b), StarGAN (Choi et al., 2018) and pix2pix (Isola et al., 2017) for this task (see Figure 3.22) and compared the performance of a recognition model trained with images generated by each method. The authors concluded that pix2pix generated more realistic/diverse LP images, as the recognition model trained with images generated by pix2pix achieved significantly better results (96.3%) than the models trained with images generated by StarGAN (94.2%) and CycleGAN (93.6%). A modified version of YOLOv2 (Redmon and Farhadi, 2017) was employed as the recognition model. As a limitation of this work, we can mention that all datasets used in the experiments are not available to the research community. Furthermore, although the authors highlighted that the more synthetic images, the better the recognition rates achieved, they created only 9k synthetic images with each GAN model without assessing at what point the recognition rates would stop increasing.

(a) Script (b) CycleGAN (c) StarGAN (d) pix2pix

Figure 3.22: Examples of Korean LPs generated by Han et al. (2020) with CycleGAN (b), StarGAN (c), and pix2pix (d). The first column shows the script images used as input (a). Image reproduced from (Han et al., 2020).

Shashirangana et al. (2022) pointed out that while there are many public datasets for ALPR, they mostly (or exclusively) have images captured during the day. As curating a new

dataset for nighttime images is both expensive and time-consuming, they employed pix2pix (Isola et al., 2017) to convert color images from the CCPD dataset (Xu et al., 2018) into thermal infrared (TIR) images. The authors explored the KAIST multi-spectral dataset (Hwang et al., 2015), which has 95k paired color and infrared images, for training the pix2pix model. Figure 3.23 shows two color images and their corresponding infrared images generated by pix2pix. As can be seen, the authors created synthetic infrared images of the entire scene and not just the LP region; hence, these images can be used to train deep models for both the detection and recognition stages. The qualitative results are promising, but experiments with public datasets were lacking to assess whether a deep model trained on such synthetic images would be able to detect/recognize LPs in real nighttime images captured by infrared cameras. In this regard, the authors mentioned a few experiments conducted on real nighttime images, but the test set had only 100 images (no samples were shown) and was not made available to the research community.



(a) Real color images (CCPD dataset) (b) Synthetic infrared images (pix2pix)

Figure 3.23: Shashirangana et al. (2022) employed pix2pix (Isola et al., 2017) to convert color images (a) into thermal infrared images (b). Image reproduced from (Shashirangana et al., 2022).

Gonçalves et al. (2019) observed that many companies and government departments do not have a large budget to invest in high-quality cameras. They also noted that forensic experts often have to handle low-quality images captured from crime scenes. Taking this into account, the authors designed a deep generative network for creating synthetic LP images as if they were acquired farther away from where they actually were. Their objective was to train a recognition model that performs better on low-resolution images (while still being robust to high-resolution images). Instead of using GANs, they employed a model very similar to a variational autoencoder (Kingma and Welling, 2014) (see Figure 3.24). They trained the model with pairs of LP images from the same vehicle, where one high-resolution image captured close to the camera is used as input, and a low-resolution image captured far from the camera is used as output. The intuition behind this training process lies in the fact that simply downscaling high-resolution images does not emulate the actual behavior of low-resolution LPs, as they contain noise resulting from long-distance captures or low-quality cameras. The experimental evaluation, carried out on images from the SSIG-ALPR dataset (Gonçalves et al., 2018), showed that adding many synthetic images (400k) of low-resolution LPs to the training set improved the recognition rate achieved by their multi-task OCR model by 4.9%. An important finding is that the accuracy on high-resolution LPs remained the same, i.e., the low-resolution samples improved the recognition model's robustness to low-resolution LPs without compromising the results obtained on high-resolution LPs. As the experiments were performed using images from a single dataset (with Brazilian LPs), it is unclear whether such a generative model needs to be retrained/adjusted for images of other LP layouts and for images acquired under other settings.

Vašek et al. (2018) explored cGAN concepts (Mirza and Osindero, 2014) to create a CNN-based super-resolution generator of LP images that converts input low-resolution images into their high-resolution counterparts closely matching the structure of the input LP patch (i.e., tilt angle, lighting conditions, among other characteristics). They trained the generator

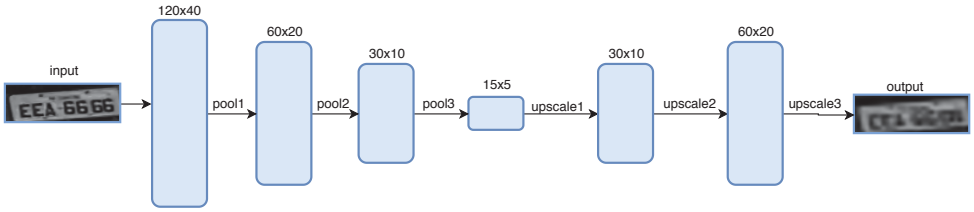


Figure 3.24: The deep generative model designed by Gonçalves et al. (2019) to create LP images simulating that they were captured farther away from where they actually were. Image reproduced from (Gonçalves et al., 2019).

using 1.6 million triplets (lr_i, hr_i, str_i), where lr_i corresponds to the input low-resolution image, hr_i refers to the desired high-resolution counterpart, and str_i is the string to be depicted on hr_i. As the focus of their work was on recognizing LPs in low-resolution videos (as detailed in Section 3.2), the authors only showed some images produced by the generator (see Figure 3.25b), without evaluating them quantitatively – i.e., without using them to train a recognition model and then assess what impact they have on its performance. It is worth noting that both the training images (1.4M real + 0.2M synthetic) and test images were taken from private datasets.

Figure 3.25: Vašek et al. (2018) proposed a super-resolution CNN-based generator that converts input low-resolution images into their high-resolution counterparts closely matching the structure of the input LP. (a) shows a simplified view of the super-resolution generator; it takes as input the low-resolution LP image and the string to be depicted. (b) shows impressive examples of high-resolution LP images created by their generator. The first column shows low-resolution images (the red strings denote the ground truth), while the second and third columns show images produced by the generator. Image adapted from (Vašek et al., 2018).

3.4 Miscellaneous

Here we present works or systems that do not fit into any of the other sections of this chapter. We first describe works where the authors designed deep models to locate the four corners of the LPs in order to rectify them before the recognition stage. We then provide information on two commercial systems that have been used frequently as baselines in the literature. Lastly, we point out some fundamental differences between scene text recognition and LPR, describing the models proposed for scene text recognition that are explored in other chapters of this work.

Meng et al. (2018) claimed that some segmentation-free methods (e.g., those proposed by Špaňhel et al. (2017) and Gonçalves et al. (2018)) might not achieve high recognition rates on considerably tilted LPs, as the respective authors only considered LPs with a regular shape and with small variations in their works. Accordingly, they designed a 10-layer CNN, called LocateNet, to predict the four vertices coordinates $(x_0/w, y_0/h, \ldots, x_3/w, y_3/h)$ of the LP. Then, an affine transformation was applied to the LP patch in order to rectify it, as illustrated in Figure 3.26. A neural network (for character segmentation) followed by AlexNet (Krizhevsky et al., 2012) (for character recognition), and three existing methods were used in their experiments

on three public datasets to demonstrate that the LP rectification stage significantly improves the recognition results. Similar findings were observed by Špaňhel et al. (2018), who also designed a deep network to locate the corners of the LP. The network, called Aligner-CNN, outputs four probability maps for the four corner points in a specified order (i.e., top left, top right, bottom right, and bottom left). The results showed that the rectification performed by Aligner-CNN considerably improved the recognition rate achieved by Holistic-CNN (Špaňhel et al., 2017) on a public dataset containing several styles of parking (e.g., parallel, angle, and perpendicular) both outside (e.g., streets, outdoor, and parking lots) and inside (e.g., parking garages). More specifically, the error rate was reduced from 12.5% to 4.0% when rectifying the LPs before recognition. However, the computational cost required for such an additional task is worth noting, as the recognition approach proposed by Meng et al. (2018) took six times longer to process an LP image compared to a baseline (Holistic-CNN) without the rectification stage (even though both methods achieved similar results). Similarly, but to a lesser extent, in (Špaňhel et al., 2018) it took about three times longer for Holistic-CNN to recognize an LP when rectifying it first.

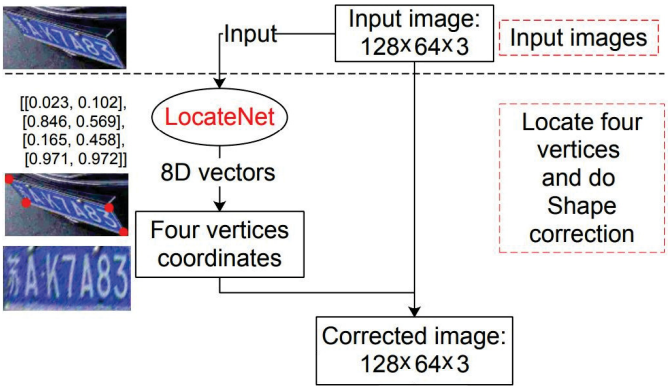


Figure 3.26: The flowchart of LocateNet (Meng et al., 2018), which predicts eight floating numbers corresponding to the horizontal and vertical locations of the four corners of the LP. Image adapted from (Meng et al., 2018).

In the same direction, Yoo and Jun (2021) evaluated five models based on deep learning to estimate the corner coordinates of tilted LP images. Considering the real-time requirements of ALPR applications, they focused on models with relatively small size and high speed. In experiments carried out on a private dataset and also on the road patrol (RP) subset of the AOLP dataset (Hsu et al., 2013), a hybrid model between a network proposed by the authors and MobileNetV2 (Sandler et al., 2018) reached the best results in terms of accuracy (i.e., the mean pixel distance between the predicted corner positions and the ground truth). In terms of efficiency, the authors compared only the sizes of the models, without detailing their execution time or the hardware used in the experiments. We believe that this is not the ideal evaluation approach, as models of similar sizes can still perform at quite different speeds due to the specific characteristics of each architecture (Huang et al., 2017; Laroca et al., 2019, 2021b).

Masood et al. (2017) presented Sighthound (Sighthound, 2024), an end-to-end ALPR system, that uses a sequence of deep CNNs for LPD, character detection (or segmentation), and character recognition. For character detection, a binary network classifier was trained with LP characters as positive examples and symbols (e.g., wheelchair, flags, among others) as negative samples. Due to its commercial nature, Sighthound's technical background is strictly confidential, i.e., little information is provided about the models used for each stage or about the datasets used to train it. According to the authors, the variety of character fonts and hard negative samples improved the robustness of their system, which outperformed other commercial solutions in two public datasets. It is worth noting that the performance of commercial systems is often

overestimated for promotional reasons (Anagnostopoulos et al., 2008; Thome et al., 2011). As it offers a trial version via an Application Programming Interface (API), Sighthound is frequently used as a baseline in the literature (Zhang et al., 2020a; Lu et al., 2021; Chen et al., 2023).

OpenALPR[9] (OpenALPR, 2024) is another commercial system often employed as a baseline in ALPR research. It offers specialized solutions for LPs from various regions, including Europe, mainland China, and the United States. This entails users inputting the correct region when using its API. While OpenALPR can deliver superior results by employing heuristic rules tailored to the specified region, the need for users to have prior knowledge regarding the LP layout can be viewed as a limitation (Laroca et al., 2021b). Indeed, studies have shown that it typically achieves better results than Sighthound on LPs from supported regions and considerably worse otherwise (Silva and Jung, 2018; Li et al., 2020; Shu et al., 2020).

ALPR is a specific application of scene text detection and recognition (Mokayed et al., 2021; Lee et al., 2022; Ding et al., 2023). Nevertheless, there are some fundamental differences between ALPR and the general task of detecting and recognizing scene text that should be highlighted: (i) there is no language model hidden in LPs, nor any substantial relationship with the context information; (ii) LPR models usually need to learn 36 character classes (10 digits [0-9], and 26 uppercase letters [A-Z]), while networks for general scene text recognition must handle 62 classes (10 digits [0-9], 26 uppercase letters [A-Z], and 26 lowercase letters [a-z]) or even more (91-96) when incorporating symbols (Shi et al., 2019; Wu et al., 2022; Jiang et al., 2023a); and (iii) detection and recognition models for ALPR do not need to deal with curved text, which is commonly encountered in natural scenes such as business logos, signs and entrances (see Figure 3.27). In the next paragraphs, we briefly describe the well-known models originally proposed for scene text recognition that are explored in other chapters of this work.

Figure 3.27: Examples of curved text, which is a commonly seen artistic-style text in natural scenes (Shi et al., 2016). Recognition models for ALPR do not need to deal with this text style. Image adapted from (Ch'ng and Chan, 2017).

Baek et al. (2019) introduced a four-stage framework (illustrated in Figure 3.28) that models the design patterns of most modern methods for scene text recognition. The *Transformation* stage removes the distortion from the word image so that the text is horizontal or normalized. This task is generally done by STNs (Jaderberg et al., 2015) with a thin-plate splines (TPS) transformation (Bookstein, 1989), which models the distortion by finding and correcting fiducial points (see the green '+' markers in Figure 3.28). 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 CNNs, such as VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2016), and Recurrent Convolutional Neural Network (RCNN) (Liang and Hu, 2015). The *Sequence Modeling* stage converts visual features to contextual features that capture the context in the sequence of characters. Bi-LSTM (Graves and Schmidhuber, 2005b,a) is generally employed for this task. Finally, the *Prediction* stage produces the character sequence

[9] Although OpenALPR has an open-source version, the commercial variant (the one typically used as a baseline) employs distinct OCR models trained with larger datasets for improved accuracy (OpenALPR, 2024).

from the identified features. This task is typically done by a CTC decoder (Graves et al., 2006) or through an attention mechanism (Bahdanau et al., 2015).

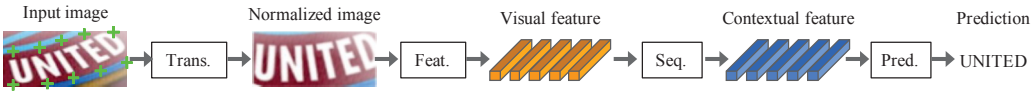


Figure 3.28: The four modules or stages of modern scene text recognition, according to (Baek et al., 2019). "Trans." stands for Transformation, "Feat." stands for Feature Extraction, "Seq." stands for Sequence Modeling, and "Pred." stands for Prediction. Image reproduced from (Baek et al., 2019).

As can be seen in Table 3.1, although most methods can fit into this framework (Atienza, 2021b), they do not necessarily have all four modules. For example, Robust text recognizer with Automatic REctification (RARE) (Shi et al., 2016), SpaTial Attention Residue Network (STAR-Net) (Liu et al., 2016), and TPS-ResNet-BiLSTM-Attention (TRBA) (Baek et al., 2019) rectify the input image using TPS, whereas CRNN (Shi et al., 2017), Recursive Recurrent neural networks with Attention Modeling (R^2AM) (Lee and Osindero, 2016), Gated Recurrent Convolution Neural Network (GRCNN) (Wang and Hu, 2017), and Rosetta (Borisyuk et al., 2018) do not normalize the input image. For the feature extraction task, RARE and CRNN use VGG; R^2AM and GRCNN employ RCNN; and STAR-Net, Rosetta and TRBA use ResNet. Regarding the sequence modeling stage, R^2AM and Rosetta skip it to speed up prediction, while RARE, STAR-Net, CRNN, GRCNN and TRBA address it using Bi-LSTMs. Lastly, R^2AM, RARE and TRBA rely on an attention mechanism to predict the sequence of characters, whereas STAR-Net, CRNN, GRCNN and Rosetta employ CTC. For more information about the methods mentioned in this paragraph, see the respective works where they were proposed and also (Atienza, 2021b; Chen et al., 2022), which summarize the similarities and differences between them.

Table 3.1: Summary of seven well-known models for scene text recognition that fit into the framework introduced by Baek et al. (2019). We list these models (and not others) as they are explored in other chapters of this work.

Model	Transformation	Feature Extraction	Sequence Modeling	Prediction
R^2AM (Lee and Osindero, 2016)	–	RCNN	–	Attention
RARE (Shi et al., 2016)	TPS	VGG	Bi-LSTM	Attention
STAR-Net (Liu et al., 2016)	TPS	ResNet	Bi-LSTM	CTC
CRNN (Shi et al., 2017)	–	VGG	Bi-LSTM	CTC
GRCNN (Wang and Hu, 2017)	–	RCNN	Bi-LSTM	CTC
Rosetta (Borisyuk et al., 2018)	–	ResNet	–	CTC
TRBA (Baek et al., 2019)	TPS	ResNet	Bi-LSTM	Attention

Inspired by the success of Vision Transformer (ViT) (Dosovitskiy et al., 2021), Atienza (2021b) proposed a simple single-stage model – called ViTSTR – that uses a pre-trained ViT (Touvron et al., 2021) to perform scene text recognition. The ViT introduced by Dosovitskiy et al. (2021) is an architecture directly inherited from Natural Language Processing (NLP) (Vaswani et al., 2017) but applied to image classification with raw image patches as input. As shown in Figure 3.29, in ViTSTR, the input image is first converted into non-overlapping patches. The patches are then converted into 1-D vector embeddings (i.e., flattened 2-D patches). As input to the encoder, a learnable patch embedding is added together with a position encoding for each embedding. ViTSTR is trained in an end-to-end manner with no parameters frozen. Considering that little emphasis has been placed on speed and computational efficiency in scene text recognition, the authors also proposed two smaller versions of ViTSTR, called ViTSTR-Tiny and ViTSTR-Small, with reduced embedding size and number of heads (see Table 3.2).

Another model that is explored in other chapters of this work and therefore should be described here is Fast-OCR (Laroca et al., 2021a). It was proposed for reading energy/gas/water

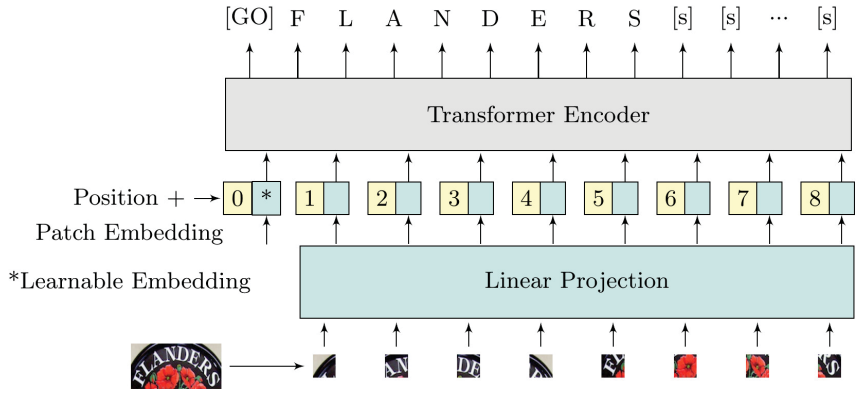


Figure 3.29: The network architecture of ViTSTR (Atienza, 2021b). The prediction head is the only difference between ViT (Dosovitskiy et al., 2021) and ViTSTR. Instead of single object-class recognition, ViTSTR must identify multiple characters with the correct sequence order and length. [GO] is a pre-defined start of sequence symbol, while [s] represents a space or end of a character sequence. Image reproduced from (Atienza, 2021b).

Table 3.2: The settings of each ViTSTR version. Table reproduced from (Atienza, 2021b).

Version	Patch Size	Depth	Embedding Size	# Heads	Sequence Length
ViTSTR-Tiny	16	12	192	3	27
ViTSTR-Small	16	12	384	6	27
ViTSTR-Base	16	12	768	12	27

meters and incorporates features from three object detection-based models focused on the speed/accuracy trade-off, namely YOLOv2 (Redmon and Farhadi, 2017), CR-NET (Silva and Jung, 2020) and Fast-YOLOv4 (Bochkovskiy, 2020). Accordingly, it is trained to predict N character classes (originally 10 classes [0-9]) using the region-of-interest patch as well as the class and bounding box (x, y, w, h) of each character as input. As detailed in Table 3.3, Fast-OCR performs detection at two different scales, as Fast-YOLOv4. The convolutional layers mostly have 3×3 kernels and the number of filters is doubled after each max-pooling layer, as in YOLOv2 and CR-NET. In addition, there are 1×1 convolutional layers between 3×3 convolutions to reduce the feature space from preceding layers. In experiments carried out on two public datasets with images of energy meters, Fast-OCR achieved considerably better results than baselines that perform recognition holistically, including CRNN, TRBA and the multi-task network designed specifically for counter recognition[10] by Gómez et al. (2018).

Table 3.3: The architecture of Fast-OCR (Laroca et al., 2021a).

#	Layer	Filters	Size	Input	Output
0	conv	32	$3 \times 3/1$	$384 \times 128 \times 3$	$384 \times 128 \times 32$
1	max		$2 \times 2/2$	$384 \times 128 \times 32$	$192 \times 64 \times 32$
2	conv	64	$3 \times 3/1$	$192 \times 64 \times 32$	$192 \times 64 \times 64$
3	max		$2 \times 2/2$	$192 \times 64 \times 64$	$96 \times 32 \times 64$
4	conv	128	$3 \times 3/1$	$96 \times 32 \times 64$	$96 \times 32 \times 128$
5	max		$2 \times 2/2$	$96 \times 32 \times 128$	$48 \times 16 \times 128$
6	conv	256	$3 \times 3/1$	$48 \times 16 \times 128$	$48 \times 16 \times 256$
7	conv	128	$1 \times 1/1$	$48 \times 16 \times 256$	$48 \times 16 \times 128$
8	conv	256	$3 \times 3/1$	$48 \times 16 \times 128$	$48 \times 16 \times 256$
9	max		$2 \times 2/2$	$48 \times 16 \times 256$	$24 \times 8 \times 256$
10	conv	512	$3 \times 3/1$	$24 \times 8 \times 256$	$24 \times 8 \times 512$
11	conv	256	$1 \times 1/1$	$24 \times 8 \times 512$	$24 \times 8 \times 256$
12	conv	512	$3 \times 3/1$	$24 \times 8 \times 256$	$24 \times 8 \times 512$
13	conv	45	$1 \times 1/1$	$24 \times 8 \times 512$	$24 \times 8 \times 45$
14	**detection**				
15	route [11]				$24 \times 8 \times 256$
16	conv	256	$1 \times 1/1$	$24 \times 8 \times 256$	$24 \times 8 \times 256$
17	upsample		$2\times$	$24 \times 8 \times 256$	$48 \times 16 \times 256$
18	route [17, 6]				$48 \times 16 \times 512$
19	conv	512	$3 \times 3/1$	$48 \times 16 \times 512$	$48 \times 16 \times 512$
20	conv	45	$1 \times 1/1$	$48 \times 16 \times 512$	$48 \times 16 \times 45$
21	**detection**				

[10] The counter is the region on each meter where the digits are displayed. Thus, in automatic meter reading, the digit recognition stage is often referred to as counter recognition (Laroca et al., 2019, 2021a; Rocha et al., 2022).

3.5 Final Remarks

Recent developments in deep learning (Bengio et al., 2021) have significantly contributed to improving many computer vision tasks, such as object detection and OCR, which directly benefit ALPR systems. Despite extensive research driven by the wide range of ALPR applications, there remains a significant gap between the performance levels reported in academic studies and those observed in real-world deployments. This gap can largely be attributed to the overly simplified setups used in most research endeavors. In the following paragraphs, we outline the primary limitations observed in the studies reviewed in this chapter.

What caught our attention right away was the lack of evaluation regarding the out-of-domain robustness of the proposed methods in most studies. Only recently have some researchers started conducting cross-dataset experiments to assess the generalizability of their methods. The prevalent approach involves training the models exclusively on the CCPD dataset and testing them on the CLPD and PKU datasets, all three acquired in mainland China (Zou et al., 2020; Wang et al., 2022c; Chen et al., 2023). We argue that it is crucial to expand such evaluations to datasets gathered from different regions, encompassing a greater diversity in LP styles. Zeni and Jung (2020) set a valuable example in this regard. They explored five datasets from various regions, three for both training and testing, and two exclusively for testing. Interestingly, both their LPR method and a baseline they trained exhibited signs of overfitting on images from unseen datasets, especially on LPs from regions with limited representation in the training data.

Even in traditional intra-dataset experiments (where training and testing data come from disjoint parts of the same dataset), it is quite common for only a particular LP style (e.g., single-row blue LPs from mainland China) to be considered in the experiments (Han et al., 2020; Maier et al., 2022; Shvai et al., 2023). To experiment with multiple LP layouts, many researchers have opted to train separate instances of their models for each layout (e.g., considering the LPR stage, one model recognizes LPs from the Taiwan region, another model recognizes LPs from mainland China, and so on) (Zhang et al., 2021d; Wang et al., 2022c; Ke et al., 2023). As one may infer, dealing with the problem in this way becomes cumbersome (even unfeasible) as the number of LP layouts the ALPR system must detect and recognize increases, since the parameters are individually adjusted for each LP layout and adding support for a new region requires retraining the networks. Moreover, this protocol does not make it possible to assess whether the proposed models, as they were designed and trained, can effectively deal with LPs from multiple regions.

To better illustrate the importance of the points discussed above, Figure 3.30 shows the predictions made by two pre-trained instances of the CR-NET model, one provided by Silva and Jung (2018) and the other by Laroca et al. (2021b), on two randomly selected images of Mercosur LPs. Although excellent recognition results were reported in these works, both models failed to correctly recognize the LPs, even though the images were free of shadows, blur, dirt, or occlusions. This suggests a potential issue with the training data. Neither study included images of vehicles bearing Mercosur LPs in their datasets. Further experiments are necessary to ascertain whether these models (and potentially others) lack robustness specifically towards LP layouts not seen during training (e.g., the models may have failed on these Mercosur LPs due to the characters' reflective films, which are absent in other layouts), or if they struggle with images captured under conditions different from those in the training set, irrespective of the LP layout. If the latter scenario holds true, the underlying reasons for this lack of robustness must be explored. Several questions arise in this context. For example, how significant is the dataset bias issue (Torralba and Efros, 2011; Tommasi et al., 2017; Hort et al., 2023) within the context of LPR? As another example, could there be issues with the protocols typically used to split public datasets into training and test sets, potentially skewing the results reported in academic research?

(Silva and Jung, 2018): Z-M7J23 **(Silva and Jung, 2018):** --M0J35
(Laroca et al., 2021b): ---7-2- **(Laroca et al., 2021b):** -----35
GT: IHM7J23 **GT:** AIH0J35

Figure 3.30: Recognition results yielded by two instances of the CR-NET model, one trained by Silva and Jung (2018) and the other by Laroca et al. (2021b), on two images of Mercosur LPs acquired by handheld cameras. For this evaluation, we used the weights provided by the respective authors in the supplementary material of each work.

Although there are many public datasets available in the literature (we managed to find nearly 40), there is still a large number of works that perform experiments exclusively on images from private datasets. Some examples are (Liu and Chang, 2019; Jin et al., 2021; Maier et al., 2022; Akoushideh et al., 2024). The use of private datasets makes it very difficult – in some cases even impossible – to make a fair comparison between results reported in different works.

When reviewing the literature, we noticed that many authors are incredibly unaware of the existence of most public datasets for ALPR. For instance, Ismail et al. (2021) asserted that AOLP (Hsu et al., 2013) was the sole publicly available dataset suitable for ALPR. Similarly, Pan et al. (2022) stated that labeled datasets for LPD and LPR are very scarce. Similar claims were made in several other works (Gao et al., 2020b; Xu et al., 2021; Ashrafee et al., 2022; Yang et al., 2023), especially when referring to datasets collected from specific geographic regions. Considering this discussion, we assert that there is a high demand for a complete review of public datasets for ALPR, describing them in detail and highlighting their distinguishing characteristics.

Such a review would shed light on less popular datasets (in terms of citations) and assist ALPR researchers in making sound choices regarding which datasets to explore in their experiments based on the target application of their algorithms. For example, CCPD (Xu et al., 2018) stands out as the most widely used dataset in existing literature, primarily due to its widespread adoption among Chinese researchers. Nevertheless, as shown in Figure 3.31, its images were heavily compressed. Therefore, CCPD may not be ideal for training and evaluating ALPR systems intended to handle less degraded images, which is often the case. Indeed, Qiao et al. (2021) observed that some images within CCPD are too blurry for the LPs to be recognized. This limitation led Silva and Jung (2022) to exclude this dataset from their LPR experiments.

Figure 3.31: Three images that illustrate the high compression ratios in the CCPD dataset. As noted by Qiao et al. (2021); Silva and Jung (2022), it is clear that the high compression ratios impair the legibility of the LPs in some cases. We show a zoomed-in version of the vehicle's LP in the bottom-right region of each image for better viewing.

Additionally, a complete listing of existing datasets would facilitate the identification of gaps in the literature caused by the lack of datasets with specific characteristics. For example,

there is no public dataset containing images of vehicles with Mercosur LPs; such a dataset would considerably assist in developing new approaches for this LP layout. We are also unaware of any dataset comprising a substantial and balanced number of images of cars and motorcycles, which would enable researchers to give equal importance to both types of vehicles and also to LPs with one and two rows of characters during experimentation (cars typically have a single-row character arrangement on their LPs, while motorcycles usually feature characters arranged in two rows). It is noticeable that motorcycles and two-row LPs have been largely overlooked in ALPR research.

Given the difficulty in collecting and labeling thousands of images of LPs from different states or provinces across a country, generative models (mostly GANs) have increasingly been employed to create synthetic LP images with diverse characteristics. These generated images have proven instrumental in reducing biases in training sets, thereby enhancing the performance of OCR models. Nevertheless, most studies have focused on unpaired image-to-image translation methods (e.g., CycleGAN) using a large number of real images for training (100k+), without addressing how similar results could be achieved with limited training data. This need for many images restricts the applicability of such methods, as numerous images are not always available for every LP layout (Han et al., 2020; Laroca et al., 2021b; Yang et al., 2023). That is probably why Wu et al. (2018); Zhang et al. (2018b, 2021c) only generated images of LPs from mainland China, which are widely available, despite carrying out experiments on LPs from other regions (United States and Taiwan). Indeed, Zhang et al. (2018b) acknowledged that the need for an abundant source of training images is the main limitation of their approach. Furthermore, we also noticed that whether a single model could effectively generate high-quality LP images from diverse regions with varying characteristics has yet to be demonstrated. While Wu et al. (2019); Fan and Zhao (2022) produced images of LPs from multiple regions, they did so by training separate models for each region. Considering these observations, there is a clear demand for developing an approach capable of generating high-quality images of LPs from various regions, even when trained with only a few hundred real images per LP layout.

Regarding the existing methods for generating synthetic data, we observed that they have been evaluated based on the results yielded by a single OCR model. For example, Wang et al. (2017); Zhang et al. (2018b); Wu et al. (2019) evaluated the efficacy of their strategies solely based on the recognition results achieved by CRNN (Shi et al., 2017), while Zhang et al. (2019b, 2021c) considered only the recognition results reached by a CNN model based on Xception (Chollet, 2017). This evaluation approach is flawed because images produced in a specific manner may benefit certain methods much more than others; in essence, a synthetic data generation method might produce images that significantly enhance the recognition results of one model but not another. This was evidenced by Laroca et al. (2019) in the context of image-based Automatic Meter Reading (AMR), where two segmentation-free approaches (including CRNN) had a much higher performance gain than the CR-NET model (Silva and Jung, 2020), which is based on YOLO, when trained with images created by a character permutation-based synthesis data generation technique (Gonçalves et al., 2018). Therefore, while there is strong evidence of improved LPR performance through such techniques, there is a lack of studies focusing on evaluating their effectiveness using outcomes from multiple OCR models with varying characteristics. Furthermore, it remains unclear whether relying solely on one method for generating synthetic data is sufficient for achieving optimal LPR results, or if significantly superior outcomes could be obtained by integrating data generated through diverse methodologies, such as images created via character permutation, rendering-based techniques, or a GAN model.

Finally, after reviewing the literature, it became evident that most research in ALPR is narrowly focused on specific tasks. For example, Al-Shemarry and Li (2020); Mokayed et al. (2021); Ding et al. (2024) exclusively addressed the LPD stage. Similarly, Xu et al. (2021);

Schirrmacher et al. (2023); Liu et al. (2024b) only dealt with the LPR stage, while Meng et al. (2018); Špaňhel et al. (2018); Yoo and Jun (2021) concentrated on corner detection and LP rectification. There is a clear need for approaches handling ALPR in an end-to-end fashion. Such approaches should be designed and evaluated considering the common challenges encountered in real-world scenarios. These challenges include efficient detection and recognition of LPs with diverse layouts, images with varying resolutions, and LPs with different numbers of characters arranged in one or two rows. In the regime where labeled data is expensive (Björklund et al., 2019; Han et al., 2020; Gao et al., 2023), these approaches should not require hundreds of thousands of real images for training and must demonstrate robustness to images captured in domains beyond those represented in the training set.

4. THE RODOSOL-ALPR DATASET

The RodoSol-ALPR dataset[11] contains 20,000 images captured by stationary cameras located at pay tolls owned by the *Rodovia do Sol* (RodoSol) concessionaire, which operated 67.5 kilometers of a highway (ES-060) in the Brazilian state of Espírito Santo for 25 years (RodoSol, 2024).

As can be seen in Figure 4.1, there are images of different types of vehicles (e.g., cars, motorcycles, buses and trucks), captured during the day and night, from distinct lanes, on clear and rainy days, and the distance from the vehicle to the camera varies slightly. All images are available in the JPG format (quality = 95) and have a resolution of 1,280 × 720 pixels.

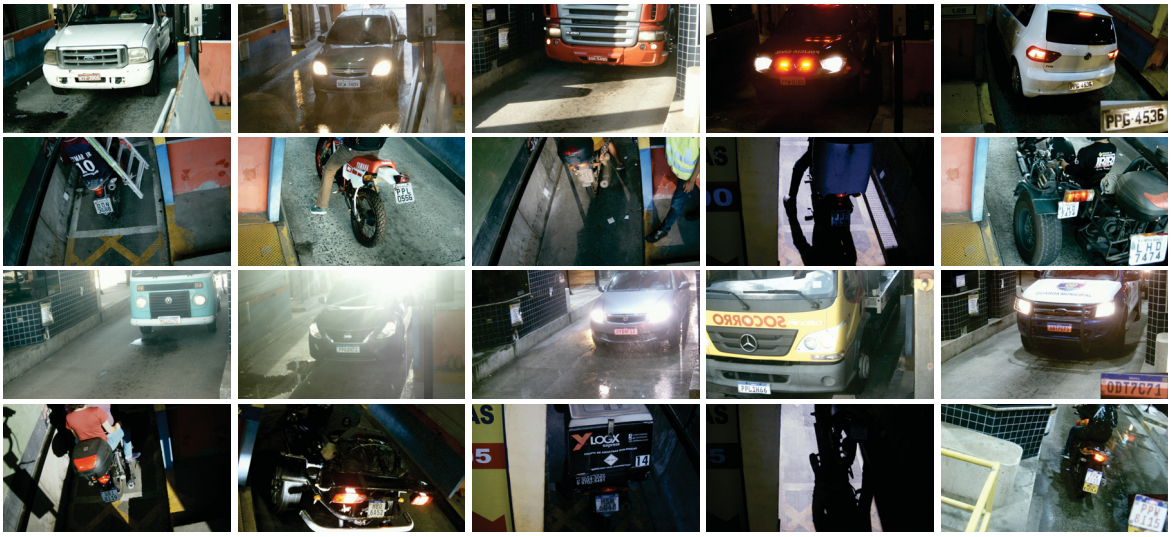


Figure 4.1: Some images extracted from the RodoSol-ALPR dataset. The first and second rows show images of cars and motorcycles, respectively, with Brazilian LPs (i.e., the standard used in Brazil before the adoption of the Mercosur standard). The third and fourth rows show images of cars and motorcycles, respectively, with Mercosur LPs. We show a zoomed-in version of the vehicle's LP in the lower right region of the images in the last column for better viewing of the LP layouts. All human faces were blurred in every image due to privacy constraints.

An important feature of this dataset is that it has images of two different LP layouts: Brazilian and Mercosur – as mentioned in Chapter 1, we use "Brazilian" to denote the layout used in Brazil before the adoption of the Mercosur layout, maintaining consistency with prior research. This feature is important because both LP layouts will coexist for many years in Brazil, as transitioning from the Brazilian to the Mercosur layout incurs costs and is not mandatory for used vehicles (Ribeiro et al., 2019; Laroca et al., 2021b). All Brazilian LPs consist of three letters followed by four digits (e.g., ABC1234), while the initial pattern adopted in Brazil for Mercosur LPs consists of three letters, one digit, one letter, and two digits (e.g., ABC1D23). In both layouts, car LPs have seven characters arranged in a single row, whereas motorcycle LPs split the characters into two rows: three on the top and four on the bottom.

Even though these two LP layouts are very similar in shape and size, there are considerable differences in their colors and characters' fonts. In Brazil, LPs have size and color variations depending on the type of the vehicle and its category (CONTRAN, 2007; MERCOSUR, 2014). In summary, car LPs have a size of 40cm × 13cm, while motorcycle LPs measure 20cm × 17cm. Private vehicles are identified by gray and black LPs in Brazilian and Mercosur

[11] The RodoSol-ALPR dataset is publicly available to the research community at `https://github.com/raysonlaroca/rodosol-alpr-dataset/`

layouts, respectively, whereas buses, taxis, and other commercial vehicles have red LPs. Further variations in color exist for specific vehicle categories, such as official or older cars. Figure 4.2 shows the diversity of the RodoSol-ALPR dataset in terms of LP characteristics.

Figure 4.2: Some LPs from the RodoSol-ALPR dataset. The first and second rows show Brazilian LPs of cars and motorcycles, respectively. The third and fourth rows show Mercosur LPs of cars and motorcycles, respectively.

We draw attention to some important characteristics of Brazilian and Mercosur LPs: (i) depending on the vehicle category, Brazilian LPs exhibit variation in both the background color and the characters' color, whereas in Mercosur LPs, only the color of the characters varies; (ii) there are Brazilian LPs with different fonts of characters (e.g., *DIN 1451 Mittelschrift* and *Mandatory*), as regulations changed over the years, whereas all Mercosur LPs have the characters printed with the *FE-Schrift* font, which contains monospaced letters and digits that are slightly disproportionate to prevent easy modification (i.e., faking one character into another) and to improve machine readability. Note that the characters '0' and 'O' are different in this font, unlike many other fonts where they look exactly the same (e.g., *Mandatory*); and (iii) in both layouts, the characters printed on motorcycle LPs are smaller in both width and height than those printed on car LPs (CONTRAN, 2007; MERCOSUR, 2014). This, coupled with the typical tilting of motorcycle LPs, inherently increases the difficulty of recognizing motorcycle LPs.

RodoSol-ALPR's 20,000 images are divided as follows: 5,000 images of cars with Brazilian LPs; 5,000 images of motorcycles with Brazilian LPs; 5,000 images of cars with Mercosur LPs; and 5,000 images of motorcycles with Mercosur LPs. As far as we know, RodoSol-ALPR is the public dataset for ALPR with the highest number of motorcycle images. The dataset is split as follows: 8,000 images for training; 8,000 images for testing; and 4,000 images for validation, following the split protocol (i.e., 40%/40%/20%) adopted in the SSIG-SegPlate (Gonçalves et al., 2016a) and UFPR-ALPR (Laroca et al., 2018) datasets. We preserved the percentage of samples for each vehicle type and LP layout; for example, there are 2,000 images of cars with Brazilian LPs in each of the training and test sets, and 1,000 images in the validation one. For reproducibility purposes, the subsets generated are explicitly available along with the proposed dataset.

Every image has the following information available in a text file: the vehicle's type (car or motorcycle), the LP's layout (Brazilian or Mercosur), its text (e.g., ABC-1234), and the position (x, y) of each of its four corners[12]. We labeled the corners instead of just the LP bounding box to enable the training of methods that explore LP rectification and the application of a wider range of data augmentation techniques.

The datasets for ALPR are generally very unbalanced in terms of character classes due to LP allocation policies (Anagnostopoulos et al., 2006; Sun et al., 2019; Zhang et al., 2021c). In Brazil, for example, one letter can appear much more often than others according to the state in

[12] We used two open source tools for labeling the dataset, namely, *sloth* and *labelImg*. They are available at `https://github.com/cvhciKIT/sloth` and `https://github.com/tzutalin/labelImg`, respectively.

which the LP was issued (Gonçalves et al., 2018; Laroca et al., 2018). This information must be taken into account when training LPR models in order to avoid undesirable biases; for instance, a network trained exclusively in the RodoSol-ALPR dataset may learn to always classify the first character as 'P' in cases where it should be 'B' or 'R' since it appears much more often in that position than these two characters (see Figure 4.3). Such biases are usually mitigated through synthetic data (Zhang et al., 2021c; Hasnat and Nakib, 2021; Shvai et al., 2023).

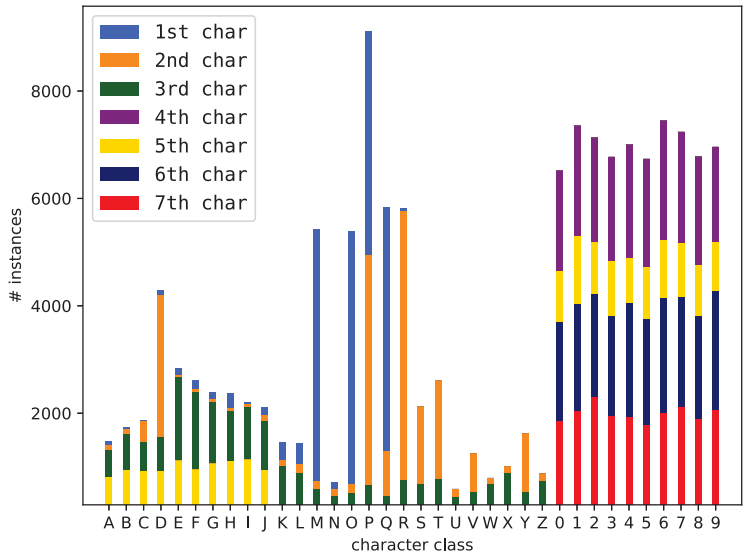


Figure 4.3: The distribution of character classes in the RodoSol-ALPR dataset. Observe that there is a significant imbalance in the distribution of the letters (due to LP allocation policies), whereas the digits are well balanced.

Regarding privacy concerns related to this dataset, we remark that in Brazil the LPs are related to the respective vehicles, i.e., no public information is available about the vehicle drivers/owners (Presidência da República, 1997; Oliveira et al., 2021). Moreover, all human faces (e.g., drivers or RodoSol's employees) were manually redacted (i.e., blurred) in each image.

5. ON THE CROSS-DATASET GENERALIZATION IN LICENSE PLATE RECOGNITION

Deep learning-based ALPR systems have often achieved recognition rates above 99% in existing datasets under the *traditional-split* protocol, where the test images mostly belong to scenarios seen during training. However, as already mentioned, in real-world applications, new cameras are regularly being installed in new locations without existing systems being retrained as often, which can dramatically decrease their performance.

Considering the above discussion, in this chapter we evaluate various OCR models in a *leave-one-dataset-out* experimental setup using nine public datasets with distinct characteristics[13]. The results obtained are compared with those achieved under the traditional-split protocol. Aligning with recent research trends (Nascimento et al., 2022, 2023; Schirrmacher et al., 2023; Liu et al., 2024b), we focus our analysis on the LPR stage. Thus, we simply train the YOLOv4 model (Bochkovskiy et al., 2020) to detect the LPs in the input images. For completeness, we also report the results achieved in the LPD stage under both of the aforementioned protocols.

In the following sections, we describe the setup adopted in our experiments. We first list the models we implemented, elucidating the rationale behind their selection over others. Afterward, we provide implementation details, including the framework used for training and testing each model, along with the associated hyperparameters. We then present and briefly describe the datasets used, as well as the techniques employed to prevent overfitting. Subsequently, we detail the evaluation protocols (traditional-split and leave-one-dataset-out), specifying which images from each dataset were used for training or testing in each experiment. Lastly, we elucidate our methodology for performance evaluation.

5.1 OCR Models

We apply 12 OCR models to LPR: RARE (Shi et al., 2016), R^2AM (Lee and Osindero, 2016), STAR-Net (Liu et al., 2016), CRNN (Shi et al., 2017), GRCNN (Wang and Hu, 2017), Holistic-CNN (Špaňhel et al., 2017), Multi-Task-LR (Gonçalves et al., 2019), Rosetta (Borisyuk et al., 2018), TRBA (Baek et al., 2019), CR-NET (Silva and Jung, 2020), Fast-OCR (Laroca et al., 2021a), and ViTSTR-Base (Atienza, 2021b). Table 5.1 presents an overview of these models, listing the original OCR application for which they were designed as well as the framework we used to train and evaluate them. We adjusted the architectures of these models to accommodate images with a width-to-height ratio of 3 at the respective input layers.

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) (Baek et al., 2019, 2021a; Atienza, 2021a,b; Nascimento et al., 2023; Dai et al., 2024). Second, we are confident in our ability to train and adjust them effectively to ensure fairness in our experiments, as the respective authors provided enough details about the model architectures, and also because we designed/employed similar networks in (Gonçalves et al., 2018, 2019; Laroca et al., 2019, 2021a).

The CR-NET and Fast-OCR models are based on the YOLO object detector (Redmon et al., 2016). Therefore, they simultaneously detect and classify the characters in the LP

[13] This chapter – in article form – was accepted for presentation at the *2022 International Conference on Computer Vision Theory and Applications (VISAPP)* (Laroca et al., 2022a). While the general conclusions remain the same, the specific recognition rates presented here differ from those in the article. This is because subsequent optimizations to the testing algorithm yielded improved performance across all OCR models, particularly for two-row LPs.

Table 5.1: OCR models explored in this chapter.

Model	Original Application
Framework: PyTorch (Atienza, 2022)	
R^2AM (Lee and Osindero, 2016)	Scene Text Recognition
RARE (Shi et al., 2016)	Scene Text Recognition
STAR-Net (Liu et al., 2016)	Scene Text Recognition
CRNN (Shi et al., 2017)	Scene Text Recognition
GRCNN (Wang and Hu, 2017)	Scene Text Recognition
Rosetta (Borisyuk et al., 2018)	Scene Text Recognition
TRBA (Baek et al., 2019)	Scene Text Recognition
ViTSTR-Base (Atienza, 2021b)	Scene Text Recognition
Framework: Keras (Chollet et al., 2024)	
Holistic-CNN (Špaňhel et al., 2017)	License Plate Recognition
Multi-Task-LR (Gonçalves et al., 2019)	License Plate Recognition
Framework: Darknet (Bochkovskiy, 2023)	
CR-NET (Silva and Jung, 2020)	License Plate Recognition
Fast-OCR (Laroca et al., 2021a)	Image-based Meter Reading

region. The networks are trained to predict 35 classes (0-9, A-Z, where 'O' and '0' are detected/recognized jointly) using the bounding box of each LP character as input. Although these models have been attaining impressive results, they require laborious data annotations, i.e., each character's bounding box needs to be labeled for training them (Zeni and Jung, 2020; Wang et al., 2022c; Liu et al., 2024b). All the other 10 models, on the other hand, output the LP characters in a segmentation-free manner, i.e., they predict the characters (also 35 classes) holistically from the LP region without the need to detect/segment each of them. Some of the models are multi-task networks, i.e., those proposed by Špaňhel et al. (2017) and Gonçalves et al. (2019) (see Section 3.2), while the others are CTC-, attention- and Transformer-based networks originally proposed for scene text recognition (see Section 3.4). According to previous works (Gonçalves et al., 2019; Hasnat and Nakib, 2021; Shvai et al., 2023), the generalizability of such segmentation-free models tends to improve significantly through the use of synthetic data.

Here we list the hyperparameters employed in each framework for training the OCR models. These hyperparameters were determined based on existing research (Baek et al., 2019; Atienza, 2021b; Oliveira et al., 2021) 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.

5.2 Datasets

Researchers have conducted experiments on various datasets to showcase the effectiveness of their models in detecting and recognizing LPs from different regions (Henry et al., 2020; Lee et al., 2022; Dai et al., 2024). Accordingly, we perform our experiments using images from the RodoSol-ALPR dataset and eight public datasets widely adopted in ALPR research (Chen et al., 2023; Ding et al., 2024; Liu et al., 2024a). These datasets are Caltech Cars (Weber, 1999), EnglishLP (Srebrić, 2003), UCSD-Stills (Dlagnekov and Belongie, 2005), ChineseLP (Zhou et al., 2012), AOLP (Hsu et al., 2013), OpenALPR-EU (OpenALPR, 2016), SSIG-SegPlate (Gonçalves

et al., 2016a), and UFPR-ALPR (Laroca et al., 2018). Table 5.2 provides an overview of these datasets, which were introduced over the past quarter-century and exhibit considerable diversity in terms of the number of images, acquisition settings, image resolution, and LP layouts.

Table 5.2: Datasets explored in this chapter. As mentioned earlier, the "Chinese" layout refers to LPs of vehicles registered in mainland China, while the "Taiwanese" layout refers to LPs of vehicles registered in the Taiwan region.

Dataset	Year	Images	Resolution	LP Layout
Caltech Cars	1999	126	896×592	American
EnglishLP	2003	509	640×480	European
UCSD-Stills	2005	291	640×480	American
ChineseLP	2012	411	Various	Chinese
AOLP	2013	2049	Various	Taiwanese
OpenALPR-EU	2016	108	Various	European
SSIG-SegPlate	2016	2000	1920×1080	Brazilian
UFPR-ALPR	2018	4500	1920×1080	Brazilian
RodoSol-ALPR	2022	20000	1280×720	Brazilian/Mercosur

Figure 5.1 highlights the variety of the chosen datasets in terms of LP layouts. It is clear that even LPs from the same country can be quite different; for example, the Caltech Cars and UCSD-Stills datasets were collected in the same region (California, United States), but they have images of LPs with significant differences in terms of colors, aspect ratios, backgrounds, and the number of characters. Additionally, the LPs may be tilted or have lower resolutions due to camera quality or vehicle-to-camera distance. It is also worth noting that some datasets (i.e., EnglishLP, UFPR-ALPR and RodoSol-ALPR) include LPs with two rows of characters.

(a) Caltech Cars (b) EnglishLP (c) UCSD-Stills (d) ChineseLP (e) AOLP (f) OpenALPR-EU (g) SSIG-SegPlate (h) UFPR-ALPR

Figure 5.1: Some representative LPs from the public datasets used in this chapter's experiments. Several LPs from the RodoSol-ALPR dataset are shown in Figure 4.2.

To mitigate biases from the public datasets, we incorporated 772 images from the internet – those labeled and provided by Laroca et al. (2021b) – into the training set. These images include 257 American LPs, 347 Chinese LPs, and 178 European LPs.

We opted not to explore the CCPD dataset (Xu et al., 2018) 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 (see Section 3.5) , significantly reducing the legibility of the LPs (Qiao et al., 2021; Silva and Jung, 2022), and this does not align with our intended application. Second, the CCPD dataset experienced multiple updates and expansions since its introduction. Hence, there is inconsistency regarding the dataset's size across different studies. While some sources claim it contains 250k images (Liang et al., 2022; Fan and Zhao, 2022; Ding

et al., 2024), others suggest a range of 280-290k images (Zou et al., 2020; Wang et al., 2022c; Gao et al., 2023), 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.

5.2.1 Synthetic Data

As shown in Table 5.2, two-thirds of the images used in our experiments are from the RodoSol-ALPR dataset. To prevent overfitting, we initially balanced the number of images from different datasets through data augmentation techniques such as random cropping, conversion to grayscale, and random perturbations of hue, saturation and brightness. We used Albumentations (Buslaev et al., 2020), a popular library mentioned in Section 2.3, to apply these transformations. Nevertheless, preliminary experiments showed that some of the OCR models were prone to predict only LP patterns present in the training set, as some patterns were being fed numerous times per epoch to the networks – particularly those belonging to smaller datasets, where many images were created from a single original one. This phenomenon was also observed in (Zhang et al., 2020c; Hasnat and Nakib, 2021; Garcia-Bordils et al., 2022).

Drawing inspiration from Gonçalves et al. (2018), we performed random permutations of character positions on each LP to mitigate potential biases during the learning phase, as depicted in Figure 5.2. As annotating bounding boxes for LP characters is a time-consuming and labor-intensive task, we chose not to explore the RodoSol-ALPR dataset for generating new images in this manner. We believe this decision is not of significant concern given the substantial size of the RodoSol-ALPR dataset compared to others. We relied on the labels provided by Laroca et al. (2021b) to explore the images from the remaining datasets.

Figure 5.2: Illustration of the character permutation-based synthetic data generation method (Gonçalves et al., 2018) we adopted to reduce overfitting. The images in rows 2 to 4 were created based on the images shown in the top row.

In this process, we do not enforce the generated LPs to have the same arrangement of letters and digits as the original LPs so that the OCR models do not memorize specific patterns from different LP layouts. For example, as described in Chapter 4, all Brazilian LPs consist of three letters followed by four digits, while Mercosur LPs in Brazil have three letters, one digit, one letter and two digits, in that order. Considering that LPs of these layouts are relatively similar, the segmentation-free networks would probably predict three letters followed by four digits for most Mercosur LPs when holding the RodoSol-ALPR dataset out in a leave-one-dataset-out evaluation, as none of the other datasets include images of vehicles with Mercosur LPs.

5.3 Evaluation Protocols

We propose a traditional-split *versus* leave-one-dataset-out experimental setup. In the following subsections (Sections 5.3.1 and 5.3.2), we describe these two protocols in detail.

5.3.1 Traditional-Split

The traditional-split protocol assesses the ability of the models to perform well in seen scenarios, as each model is trained on the union of the training set images from all datasets and evaluated on the test set images from the respective datasets. In recent works, researchers have chosen to train a single model on images from multiple datasets (instead of training a specific network for each dataset or LP layout, as was commonly done in the past) so that the proposed models are robust for different scenarios with considerably less manual effort since their parameters are adjusted only once for all datasets (Selmi et al., 2020; Qin and Liu, 2022; Silva and Jung, 2022).

For reproducibility, it is important to make clear how we divided the images from each of the datasets to train, validate and test the chosen models[14]. The UCSD-Stills, SSIG-SegPlate, UFPR-ALPR and RodoSol-ALPR datasets were split according to the protocols defined by the respective authors, while the other datasets – which do not have well-defined evaluation protocols – were divided following previous works. In summary, as in (Xiang et al., 2019; Henry et al., 2020; Liu et al., 2024a), the Caltech Cars dataset was randomly split into 63.5% of the images for training/validation and 36.5% for testing. Following (Panahi and Gholampour, 2017; Beratoğlu and Töreyin, 2021), the EnglishLP dataset was randomly divided as follows: 80% of the images for training/validation and 20% for testing. For the ChineseLP dataset, we employed the same protocol as in our previous work (Laroca et al., 2021b): 40% of the images for training, 20% for validation, and 40% for testing. We split 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, following (Xie et al., 2018; Zhuang et al., 2018; Liang et al., 2022), with 20% of the training images being used for validation. Finally, as in most works in the literature (Masood et al., 2017; Xu et al., 2022; Zibani et al., 2024), we used all 108 images from the OpenALPR-EU dataset for testing (this division has been considered as a mini leave-one-dataset-out evaluation in recent works). Table 5.3 lists the exact number of images used for training, validating and testing the chosen models.

Table 5.3: A summary of each dataset's image distribution across training, validation, and test sets.

Dataset	Training	Validation	Test	Discarded	Total
Caltech Cars	61	16	46	3	126
EnglishLP	326	81	102	0	509
UCSD-Stills	181	39	60	11	291
ChineseLP	159	79	159	14	411
AOLP	1,093	273	683	0	2,049
OpenALPR-EU	0	0	108	0	108
SSIG-SegPlate	789	407	804	0	2,000
UFPR-ALPR	1,800	900	1,800	0	4,500
RodoSol-ALPR	8,000	4,000	8,000	0	20,000

As indicated in Table 5.3, a small fraction of the images (0.01%) was excluded from our experiments[15], either because it is impossible to recognize the LPs on them due to occlusion, lighting or image acquisition problems, or because they do not represent real ALPR scenarios (e.g., images showing a person holding an LP). Figure 5.3 shows three illustrative examples. Such images were also discarded in previous works (Masood et al., 2017; Laroca et al., 2021b).

[14] The complete lists of which images from each dataset were used for training, testing and validation can be downloaded at `https://raysonlaroca.github.io/supp/visapp2022/splits.zip`

[15] The complete list of discarded images can be found at `https://raysonlaroca.github.io/supp/visapp2022/discarded-images.txt`

Figure 5.3: Examples of images discarded in our experiments. Image reproduced from (Laroca et al., 2021b).

5.3.2 Leave-One-Dataset-Out

The leave-one-dataset-out protocol evaluates the generalization performance of the trained models by testing them on the test set of an independent dataset, meaning no images from that dataset are available during training. In each experiment, one dataset's test set becomes the unseen data, while the models are trained on all images from the remaining datasets. For example, if the test set from UCSD-Stills is the current unseen data, the models are trained using all images from the Caltech Cars, EnglishLP, ChineseLP, AOLP, OpenALPR-EU, SSIG-SegPlate, UFPR-ALPR and RodoSol-ALPR datasets, along with the internet images labeled by Laroca et al. (2021b).

We assess the models exclusively on the test set images from each unseen dataset, without incorporating the training and validation images in the assessment. This ensures that the results achieved by each model on a particular dataset remain entirely comparable to those obtained by the same model under the traditional-split protocol. For clarity, we illustrate in Figure 5.4 the methodology used for conducting the experiments under the leave-one-dataset-out protocol.

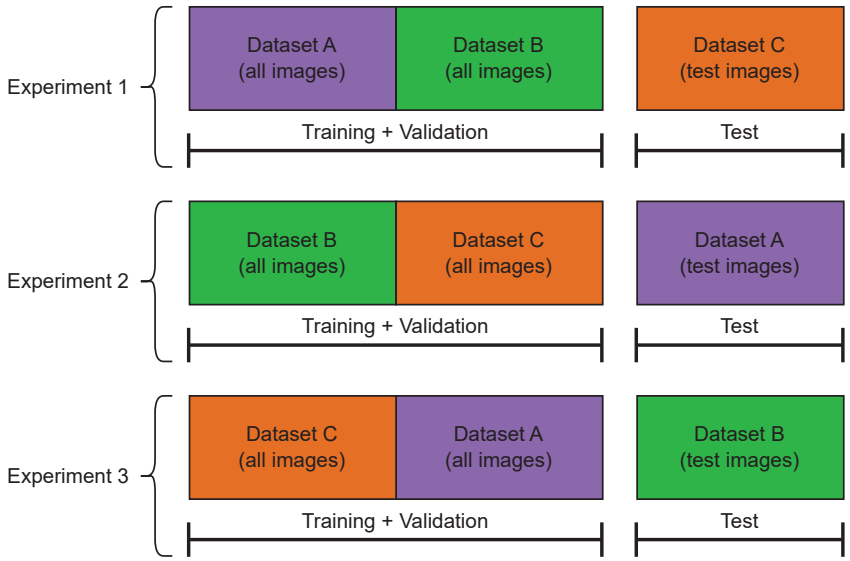


Figure 5.4: An illustration of how the experiments are conducted under the leave-one-dataset-out protocol. Here, only three datasets are considered for simplicity.

5.4 Performance Evaluation

The LP regions fed into the OCR models were detected using YOLOv4 (Bochkovskiy et al., 2020) – with an input size of 672×416 pixels – rather than being directly cropped from the ground truth. 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 OCR models when faced

with imprecisely detected LP regions (Gonçalves et al., 2018; Lee et al., 2022). We opted for YOLOv4 because YOLO-based models have consistently achieved impressive results in ALPR research (Weihong and Jiaoyang, 2020; Laroca et al., 2021b; Yang et al., 2023) (at the time of conducting the experiments for this chapter, YOLOv4 was the latest model available). As detailed in the next section, YOLOv4 reached an average recall rate exceeding 99.5% in our experiments, considering detections with Intersection over Union (IoU) ≥ 0.5 with the ground truth as correct.

As mentioned in Chapter 3, the first character in Chinese LPs is a Chinese character that represents the province in which the vehicle is affiliated. Even though Chinese LPs are used in our experiments (see Figure 5.1d), the models were not trained or adjusted to recognize Chinese characters; that is, only digits and English letters are considered. This same procedure has been adopted in many works (Selmi et al., 2020; Laroca et al., 2021b; Shashirangana et al., 2022) for several reasons, including scope reduction and the fact that it is not trivial for non-Chinese speakers to analyze the different Chinese characters in order to make an accurate error analysis or to choose which synthetic data generation techniques to explore. Accordingly, even Chinese authors have reported the recognition rates achieved by their methods when considering only digits and English letters (Wu et al., 2018; Zhang et al., 2020a,b; Fan and Zhao, 2022; Chen et al., 2023). Following (Li et al., 2019), we denoted all Chinese characters as a single class '*' in our experiments. Our results demonstrate that the models effectively learned to distinguish between Chinese characters and other characters (digits and English letters), with this approach minimally impacting the recognition of non-Chinese characters.

All metrics reported in our experiments were described in Section 2.1.

5.5 Results and Discussion

First, we report in Table 5.4 the recall rates obtained by the YOLOv4 model in the LPD stage. As can be seen, it reached surprisingly good results in both protocols. More specifically, recall rates above 99.9% were achieved in 14 of the 18 assessments. Consistent with previous works (Laroca et al., 2018; Gonçalves et al., 2018; Silva and Jung, 2020; Ding et al., 2023), the detection results are slightly worse for the UFPR-ALPR dataset due to its challenging nature, as (i) it has images where the vehicles are considerably far from the camera; (ii) some of its frames have motion blur because the dataset was recorded in real-world scenarios where both the vehicle and the camera – inside another vehicle – are moving; and (iii) it also contains images of motorcycles, where the backgrounds can be much more complicated due to different body configurations and mixtures with other background scenes (Hsu et al., 2015; Serajeh, 2016).

Table 5.4: Recall rates obtained by YOLOv4 in the LPD stage. "Trad." stands for traditional-split and "LODO" stands for leave-one-dataset-out. The number of LPs in each dataset's test set is listed below its name.

Model \ Test set	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	OpenALPR-EU # 108	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
YOLOv4 (Trad.)	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.9%	99.1%	100.0%	99.9%
YOLOv4 (LODO)	100.0%	100.0%	100.0%	100.0%	99.9%	99.1%	100.0%	96.8%	99.6%	99.5%

The precision rates achieved in our experiments were approximately 98% and 95% under the traditional-split and leave-one-dataset-out protocols, respectively. We omit a per-dataset breakdown of precision because the "false positives" identified by YOLOv4 primarily correspond to unlabeled LPs in the image backgrounds, not actual errors.

Given the results obtained under the leave-one-dataset-out protocol, we assert that deep learning models trained on a variety of datasets can be reliably applied to detect LPs in images from unseen datasets. Of course, this may not hold true in extraordinary cases where the test set

domain is very different from that of the training set, but this was not the case in our experiments carried out on images from nine datasets with diverse characteristics.

Regarding the LPR stage, the results achieved by all OCR models under the traditional-split and leave-one-dataset-out protocols are shown in Table 5.5 and Table 5.6, respectively. Table 5.6 also includes the results obtained by the Sighthound (2022) and OpenALPR (2022) commercial systems since, in principle, they are trained on images from large-scale private datasets rather than the public datasets explored here (thus aligning with the leave-one-dataset-out protocol). For further details on both systems, refer to Section 3.4.

Table 5.5: Recognition rates obtained by all models under the traditional-split protocol, which assesses the ability of the models to perform well in seen scenarios. Each model (rows) was trained once on the union of the training set images from all datasets and evaluated on the respective test sets (columns). The models are listed alphabetically, and the best recognition rate achieved in each dataset is shown in bold.

Model \ Test set # LPs	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	OpenALPR-EU # 108	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
CR-NET (Silva and Jung, 2020)	**97.8%**	94.1%	**100.0%**	**97.5%**	98.0%	96.3%	**97.5%**	82.6%	59.0%[†]	91.4%
CRNN (Shi et al., 2017)	93.5%	88.2%	91.7%	90.7%	97.1%	93.5%	92.9%	68.9%	73.6%	87.8%
Fast-OCR (Laroca et al., 2021a)	93.5%	**97.1%**	**100.0%**	**97.5%**	98.1%	**97.2%**	97.1%	81.6%	56.7%[†]	91.0%
GRCNN (Wang and Hu, 2017)	93.5%	92.2%	93.3%	91.9%	97.1%	87.0%	93.4%	66.6%	77.6%	88.1%
Holistic-CNN (Špaňhel et al., 2017)	87.0%	75.5%	88.3%	95.0%	97.7%	89.8%	95.6%	81.2%	94.7%	89.4%
Multi-Task-LR (Gonçalves et al., 2019)	89.1%	73.5%	85.0%	92.5%	94.9%	85.2%	93.3%	72.3%	86.6%	85.8%
R^2AM (Lee and Osindero, 2016)	89.1%	83.3%	86.7%	91.9%	96.5%	88.9%	92.0%	75.9%	83.4%	87.5%
RARE (Shi et al., 2016)	95.7%	94.1%	95.0%	94.4%	97.7%	94.4%	94.0%	75.7%	78.7%	91.1%
Rosetta (Borisyuk et al., 2018)	89.1%	82.4%	93.3%	93.8%	97.5%	90.7%	94.4%	75.5%	89.0%	89.5%
STAR-Net (Liu et al., 2016)	95.7%	96.1%	95.0%	95.7%	97.8%	**97.2%**	96.1%	78.8%	82.3%	**92.7%**
TRBA (Baek et al., 2019)	93.5%	91.2%	91.7%	93.8%	97.2%	93.5%	97.3%	83.4%	80.6%	91.3%
ViTSTR-Base (Atienza, 2021b)	87.0%	88.2%	86.7%	96.9%	**99.4%**	89.8%	95.8%	**89.7%**	**95.6%**	92.1%
Average	92.0%	88.0%	92.2%	94.3%	97.4%	92.0%	95.0%	77.7%	79.8%	89.8%

[†] 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 (as detailed in Section 5.1).

Table 5.6: Recognition rates obtained by all models under the leave-one-dataset-out protocol, which assesses the generalization performance of the models by testing them on the test set images of an unseen dataset. For each dataset (columns), we trained the models (rows) on all images from the other datasets. The models are listed alphabetically, and the best recognition rates achieved are shown in bold.

Approach \ Test set	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	OpenALPR-EU # 108	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
CR-NET (Silva and Jung, 2020)	**97.8%**	**97.1%**	**98.3%**	94.4%	**89.1%**	**98.1%**	97.1%	66.4%	**63.8%**	**89.1%**
CRNN (Shi et al., 2017)	93.5%	82.4%	86.7%	84.5%	71.6%	94.4%	90.8%	62.9%	39.2%	78.4%
Fast-OCR (Laroca et al., 2021a)	95.7%	95.1%	96.7%	93.8%	79.3%	96.3%	95.5%	65.9%	63.4%	86.8%
GRCNN (Wang and Hu, 2017)	93.5%	82.4%	93.3%	85.1%	72.1%	91.7%	90.8%	62.7%	40.0%	79.0%
Holistic-CNN (Špaňhel et al., 2017)	84.8%	56.9%	76.7%	82.6%	60.0%	93.5%	93.2%	66.4%	34.5%	72.0%
Multi-Task-LR (Gonçalves et al., 2019)	84.8%	57.8%	78.3%	76.4%	67.5%	88.9%	90.8%	61.7%	25.2%	70.2%
R^2AM (Lee and Osindero, 2016)	89.1%	58.8%	81.7%	85.1%	62.6%	89.8%	94.2%	61.2%	41.1%	73.7%
RARE (Shi et al., 2016)	89.1%	64.7%	93.3%	88.2%	70.7%	92.6%	93.9%	78.2%	40.2%	79.0%
Rosetta (Borisyuk et al., 2018)	95.7%	82.4%	88.3%	87.6%	70.6%	90.7%	93.9%	69.2%	42.8%	80.1%
STAR-Net (Liu et al., 2016)	91.3%	85.3%	93.3%	92.5%	79.2%	96.3%	93.8%	74.8%	43.8%	83.4%
TRBA (Baek et al., 2019)	91.3%	62.7%	95.0%	92.5%	75.3%	92.6%	96.8%	82.9%	42.9%	81.3%
ViTSTR-Base (Atienza, 2021b)	93.5%	62.7%	86.7%	**96.3%**	68.9%	91.7%	**97.8%**	**84.7%**	59.7%	82.4%
Average	91.7%	74.0%	89.0%	88.3%	72.2%	93.1%	94.0%	69.7%	44.7%	79.6%
Average (traditional-split protocol)	92.0%	88.0%	92.2%	94.3%	97.4%	92.0%[‡]	95.0%	77.7%	79.8%	89.8%
Sighthound (2022)	87.0%	94.1%	90.0%	84.5%	79.6%	94.4%	79.2%	52.6%	51.0%	79.2%
OpenALPR (2022)[*]	95.7%	99.0%	96.7%	93.8%	81.1%	99.1%	91.4%	87.8%	70.0%	90.5%

[‡] Even under the traditional-split protocol, no images from the OpenALPR-EU dataset were used for training. This is the protocol commonly adopted in the literature (Silva and Jung, 2018; Zibani et al., 2024).
[*] OpenALPR contains specialized solutions for LPs from different regions and we must enter the correct region before using its API. Hence, it was expected to achieve better results than the other methods.

The first observation is that, as expected, the best results – on average for all models – were attained when training and evaluating the models on disjoint subsets from the same datasets (i.e., under the traditional-split protocol). The sole exception was precisely in the OpenALPR-EU dataset, which has no training images even under the traditional-split protocol. Despite this seeming somewhat counterintuitive, we kept this division for two main reasons: (i) to maintain consistency with previous works (Silva and Jung, 2018; Xu et al., 2022; Zibani et al., 2024), which used all images from OpenALPR-EU for testing; and (ii) to analyze how the models perform when trained with additional data from other datasets, which in this case corresponds to the leave-one-dataset-out protocol since it employs all images from the other datasets – not just the training set ones – for training. While it has been acknowledged for many years that incorporating

images from other datasets into the training set may result in declines in performance (Torralba and Efros, 2011; Khosla et al., 2012), the recognition rates reached in the OpenALPR-EU dataset generally improved with more images from other datasets integrated into the training set (i.e., under the leave-one-dataset-out protocol). This enhancement is likely due to the utilization of all images from the EnglishLP dataset for training, as both datasets contain images of European LPs.

The average recognition rate across all datasets decreased from 89.8% under the traditional-split protocol to 79.6% under the leave-one-dataset-out protocol. This drastic performance drop is accentuated by the poor results obtained on the EnglishLP, AOLP and RodoSol-ALPR datasets under the leave-one-dataset-out protocol. For instance, the average recognition rate for the AOLP dataset went from 97.4% (traditional-split) to 72.2% (leave-one-dataset-out). Similarly, the average recognition rate for the RodoSol-ALPR dataset plummeted from 79.8% (traditional-split) to 44.7% (leave-one-dataset-out).

We expected such a severe drop in the recognition rates for the RodoSol-ALPR dataset, as no other dataset has images of Mercosur LPs or as many images of two-row LPs. However, we were surprised by the poor outcomes observed in the EnglishLP and AOLP datasets. Previous works have often reported recognition rates around 97% for the EnglishLP dataset and above 99% for the AOLP dataset (Henry et al., 2020; Laroca et al., 2021b; Zhang et al., 2021d; Wang et al., 2022c; Ke et al., 2023). Upon analysis, we found that most recognition errors under the leave-one-dataset-out protocol were not due to challenging scenarios but rather stemmed from differences in the fonts of the LP characters between training and test images, as well as because of specific patterns within the LPs (e.g., a coat of arms between the LP characters or a straight line below them). To better illustrate, Figure 5.5 (top row) shows four LPs from the AOLP dataset where the ViTSTR-Base model, which performed best on that dataset (99.4%), recognized at least one character incorrectly under the leave-one-dataset-out protocol but not under the traditional split. Similarly, Figure 5.5 (bottom row) shows the predictions made by STAR-Net for four LPs from the EnglishLP dataset (although STAR-Net ranked second in recognition performance on EnglishLP (96.1%), we selected it for illustration because it experienced a larger performance drop under the leave-one-dataset-out protocol than the top-performing model). These findings highlight the importance of conducting cross-dataset experiments in the ALPR context.

Figure 5.5: Comparison of the predictions yielded for the same LPs under the leave-one-dataset-out (LODO) and traditional-split (Trad.) protocols. The top row shows the predictions returned by ViTSTR-Base for four LPs from the AOLP dataset, while the bottom row shows the predictions made by STAR-Net for four LPs from EnglishLP. In general, the errors under the LODO protocol (outlined in red) were not observed in challenging cases (e.g., blurry or extremely tilted images). This suggests that these errors likely stemmed from differences between the training and testing data distributions.

The second observation is that, regardless of the evaluation protocol adopted, no OCR model achieved the best result across all datasets. Interestingly, STAR-Net attained the highest average recognition rate under the traditional-split protocol (92.7%) without securing the top

spot in eight of the nine datasets. These results emphasize the importance of evaluating the models on multiple datasets with varying characteristics, including LPs from different regions.

The third observation is that all the 12 OCR models trained by us, as well as both commercial systems, failed to reach recognition rates above 70% in the RodoSol-ALPR's test set under the leave-one-dataset-out protocol. These underwhelming results are primarily attributed to the unique composition of the RodoSol-ALPR dataset, which includes a substantial number of images featuring Mercosur LPs, motorcycles, and two-row LPs. To illustrate, OpenALPR accurately recognized 3,561 of the 4,000 Brazilian LPs in the test set (89.0%), yet only 2,039 out of the 4,000 Mercosur LPs (51.0%). Similarly, OpenALPR correctly identified 3,772 of the 4,000 car/single-row LPs in the test set (94.3%) but only 1,827 out of the 4,000 motorcycle/two-row LPs (45.7%). These results emphasize the importance of the RodoSol-ALPR dataset for the reliable recognition of Mercosur LPs and also for the accurate evaluation of ALPR systems, as it mitigates bias during assessments by incorporating an equal number of "easy" samples (cars with single-row LPs) and "difficult" samples (motorcycles with two-row LPs).

We also did not rule out challenging images when selecting the images for the creation of the RodoSol-ALPR dataset. Figure 5.6 shows some of these images along with the predictions returned by ViTSTR-Base (traditional-split) and OpenALPR, which are the top-performing model and commercial system on this dataset, respectively. The results are in line with what was stated by Zhang et al. (2021c); Lee et al. (2022); Ke et al. (2023), specifically, that there is still significant room for improvement in detecting and recognizing LPs in complex environments.

Figure 5.6: Some LP images from the RodoSol-ALPR dataset along with the predictions returned by ViTSTR-Base and OpenALPR. Observe that one character may become very similar to another due to factors such as blur, dirt, exposure levels (either too low or too high), rotations and occlusions. For correctness, we checked if the Ground Truth (GT) matched the vehicle make and model on the National Traffic Department of Brazil (DENATRAN) database.

Lastly, it is important to highlight the number of experiments we conducted for this traditional-split *versus* leave-one-dataset-out evaluation. We trained each of the 12 chosen OCR models 10 times: once following the split protocols traditionally adopted in the literature (see Table 5.5) and nine for the leave-one-dataset-out evaluation (see Table 5.6); not to mention the experiments with YOLOv4 related to the LPD stage. We remark that a single training process of some models (e.g., TRBA and ViTSTR-Base) took several days to complete on an NVIDIA Quadro RTX 8000 GPU, which is one of the best GPUs available on the market. This extensive set of experiments likely explains why a leave-one-dataset-out evaluation has not yet been conducted in the existing literature.

5.6 Final Remarks

Considering that the performance of ALPR systems under the traditional-split protocol is rapidly improving, researchers should pay more attention to cross-dataset setups. These setups better mimic real-world ALPR applications, where new cameras are frequently being installed in diverse locations without the need to retrain existing systems for each installation.

As a first step toward that direction, in this chapter we evaluated 12 OCR models on nine public datasets with a great variety in several aspects (e.g., acquisition settings, image resolution, and LP layouts). We adopted a traditional-split *versus* leave-one-dataset-out experimental setup to empirically assess the cross-dataset generalizability of the chosen models.

The experimental results showed significant drops in performance for most datasets when training and testing the OCR models in a leave-one-dataset-out fashion. The fact that very low recognition rates (around 73%) were reached in both the EnglishLP and AOLP datasets underscores the importance of carrying out cross-dataset experiments, as very high recognition rates (around 97% and 99%, respectively) have frequently been achieved on these datasets under the traditional-split protocol (Henry et al., 2020; Al-batat et al., 2022; Ke et al., 2023). Furthermore, the results accentuated the importance of the RodoSol-ALPR dataset for the robust recognition of Mercosur and two-row LPs, as all 12 models trained by us failed to reach recognition rates above 70% on its test set under the leave-one-dataset-out protocol.

Our experiments also emphasized the importance of evaluating OCR models on multiple datasets with varying characteristics, as no model emerged as superior across all datasets. In this sense, we remark that much of the current research relies on only three or fewer datasets in the experiments or concentrates solely on datasets from a specific region. Although recent studies by Laroca et al. (2021b); Lee et al. (2022); Chen et al. (2023) indicate a positive trend towards more comprehensive evaluations, progress in this direction has been relatively gradual.

Another finding that should be highlighted relates to using the YOLOv4 model in the LPD stage. YOLOv4 achieved remarkably good results under both protocols. This leads us to conclude that well-established object detectors trained on a variety of datasets can be reliably employed for LPD, even when presented with images from unseen datasets.

6. LEVERAGING MODEL FUSION FOR IMPROVED LICENSE PLATE RECOGNITION

Multiple studies, including our own presented in the previous chapter, have shown that different models exhibit varying levels of robustness across different datasets (Zeni and Jung, 2020; Mokayed et al., 2021). Each dataset poses distinct challenges, such as diverse LP layouts and varying tilt ranges. As a result, a model that performs optimally on one dataset may yield poor results on another. This raises an important question: *"Can we substantially enhance LPR results by fusing the outputs of diverse OCR 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 (Izidio et al., 2020; Nascimento et al., 2022, 2023; Schirrmacher et al., 2023). These scenarios can greatly benefit from the fusion of multiple OCR models.

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. (2020) 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 recent work by Schirrmacher et al. (2023), 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.

Taking this into account, in this chapter, we thoroughly examine the potential of enhancing LPR results through the fusion of outputs from multiple OCR models[16]. 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 chapter has two main contributions:

- We present empirical 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

[16] This chapter, in the form of an article, was accepted for presentation at the *2023 Iberoamerican Congress on Pattern Recognition (CIARP)* (Laroca et al., 2023b).

combine 4-6 fast models. Although these models may not achieve the highest accuracy individually, their fusion results in an optimal trade-off between speed and accuracy.

The remainder of this chapter is organized as follows. Section 6.1 provides an overview of the experimental setup. Subsequently, Section 6.2 delves into the presentation and analysis of the results obtained. Finally, Section 6.3 summarizes our findings.

6.1 Experimental Setup

This section provides an overview of the experimental setup adopted in this chapter. Initially, we list the models implemented, omitting detailed descriptions since they are the same ones used in the preceding chapter. Subsequently, we compile a list of the datasets employed in our assessments, showcasing sample LP images from each dataset to highlight their diversity. Finally, we elaborate on the strategies examined for fusing the outputs of the different models.

The experiments were conducted on a computer with an AMD Ryzen Threadripper 1920X 3.5GHz CPU, 96 GB of RAM operating at 2,133 MHz, an NVMe SSD (read: 3,500 MB/s; write: 3,000 MB/s), and an NVIDIA Quadro RTX 8000 GPU (48 GB).

6.1.1 OCR Models

For this study, we explored the same models used in the preceding chapter: RARE (Shi et al., 2016), R^2AM (Lee and Osindero, 2016), STAR-Net (Liu et al., 2016), CRNN (Shi et al., 2017), GRCNN (Wang and Hu, 2017), Holistic-CNN (Špaňhel et al., 2017), Multi-Task-LR (Gonçalves et al., 2019), Rosetta (Borisyuk et al., 2018), TRBA (Baek et al., 2019), CR-NET (Silva and Jung, 2020), Fast-OCR (Laroca et al., 2021a) and ViTSTR-Base (Atienza, 2021b). We chose these models not only for the reasons outlined in Section 5.1, but also because they have often served as benchmarks in LPR research (Gong et al., 2022; Chen et al., 2023; Dai et al., 2024).

As detailed in Section 5.1, we implemented each model using the original framework or well-known public repositories associated with it.

6.1.2 Datasets

As shown in Table 6.1, we have incorporated three new datasets into the collection of datasets explored in the preceding chapter (see Section 5.2). These datasets are PKU (Yuan et al., 2017), CD-HARD (Silva and Jung, 2018), and CLPD (Zhang et al., 2021c). They are popular choices for cross-dataset experiments (Fan and Zhao, 2022; Silva and Jung, 2022; Chen et al., 2023).

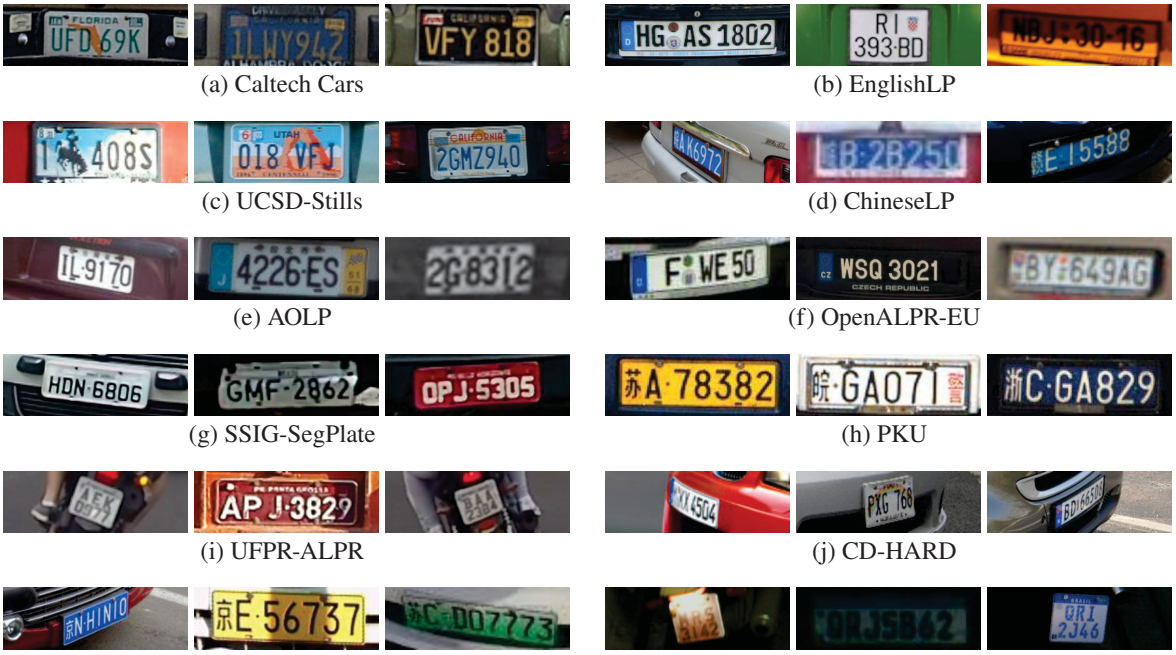
Table 6.1: The 12 datasets employed in this chapter's experiments, with * indicating those used exclusively for testing (i.e., in cross-dataset experiments). The datasets marked with "(new)" were not explored in the previous chapter.

Dataset	Year	Images	LP Layout	Dataset	Year	Images	LP Layout
Caltech Cars	1999	126	American	SSIG-SegPlate	2016	2,000	Brazilian
EnglishLP	2003	509	European	PKU* (new)	2017	2,253	Chinese
UCSD-Stills	2005	291	American	UFPR-ALPR	2018	4,500	Brazilian
ChineseLP	2012	411	Chinese	CD-HARD* (new)	2018	102	Various
AOLP	2013	2,049	Taiwanese	CLPD* (new)	2021	1,200	Chinese
OpenALPR-EU*	2016	108	European	RodoSol-ALPR	2022	20,000	Brazilian & Mercosur

Each dataset was divided using standard splits, defined by the datasets' authors, or following previous works (Laroca et al., 2021b; Wang et al., 2022c; Ke et al., 2023) in cases

where no standard split was available[17]. Specifically, eight datasets were used for both training and evaluating the models, mirroring the datasets employed in this way in the preceding chapter under the traditional-split protocol. Meanwhile, four datasets were exclusively reserved for testing purposes, comprising OpenALPR-EU along with the three newly incorporated datasets. The selected datasets exhibit substantial diversity in terms of image number, 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.

The diversity of LP layouts across the selected datasets is depicted in Figure 6.1, 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.



(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 6.1: Some LP images from the public datasets used in this chapter's experimental evaluation.

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 OCR models to memorize sequence patterns encountered during training (Zeni and Jung, 2020; Garcia-Bordils et al., 2022), we generated many synthetic LP images by shuffling the character positions on each LP (Gonçalves et al., 2018). Examples of these generated images are shown in Figure 6.2.

6.1.3 Fusion Approaches

We examine three primary approaches to combine the outputs of multiple OCR models. The first approach involves selecting the sequence predicted with the *Highest Confidence* (HC) value as the

[17] 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/`

Figure 6.2: 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.

final prediction, even if only one model predicts it. The second approach employs the *Majority Vote* (MV) rule to aggregate the sequences predicted by different models. In other words, the final prediction is 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 OCR 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 prediction made with the highest confidence among the tied predictions as the final one, and (ii) selecting the prediction made by the "best model" as the final prediction. In this study, for simplicity, we consider the best model the one that performs best individually across all datasets. However, in a more practical context, 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 use the acronym MV-HC to denote 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 approaches follow a similar naming convention (MVCP-HC and MVCP-BM).

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.

6.2 Results and Discussion

Following the methodology detailed in the previous chapter, we employed the YOLOv4 model (Bochkovskiy et al., 2020) to detect the LPs for subsequent processing by the OCR models. Considering the detections with an IoU ≥ 0.7 with the ground truth as correct, YOLOv4 achieved an average recall rate exceeding 99.7% in the intra-dataset experiments and 97.8% in the cross-dataset experiments. In both cases, the precision rates obtained were higher than 97%.

Table 6.2 shows the recognition rates obtained on the disjoint test sets of the eight datasets used for training and validating the models (intra-dataset experiments). It presents the results reached by each model individually, as well as the outcomes achieved through the fusion strategies outlined in Section 6.1.3. To improve clarity, Table 6.2 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 OCR models, refer to Table 6.3. 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 6.2: Comparison of the recognition rates achieved across eight popular datasets by 12 models individually and through five different fusion strategies (intra-dataset experiments). 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.

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
CR-NET	**97.8%**	94.1%	**100.0%**	**97.5%**	98.1%	**97.5%**	82.6%	59.0%†	90.8%
CRNN	93.5%	88.2%	91.7%	90.7%	97.1%	92.9%	68.9%	73.6%	87.1%
Fast-OCR	93.5%	**97.1%**	**100.0%**	**97.5%**	98.1%	97.1%	81.6%	56.7%†	90.2%
GRCNN	93.5%	92.2%	93.3%	91.9%	97.1%	93.4%	66.6%	77.6%	88.2%
Holistic-CNN	87.0%	75.5%	88.3%	95.0%	97.7%	95.6%	81.2%	94.7%	89.4%
Multi-Task-LR	89.1%	73.5%	85.0%	92.5%	94.9%	93.3%	72.3%	86.6%	85.9%
R^2AM	89.1%	83.3%	86.7%	91.9%	96.5%	92.0%	75.9%	83.4%	87.4%
RARE	95.7%	94.1%	95.0%	94.4%	97.7%	94.0%	75.7%	78.7%	90.7%
Rosetta	89.1%	82.4%	93.3%	93.8%	97.5%	94.4%	75.5%	89.0%	89.4%
STAR-Net	95.7%	96.1%	95.0%	95.7%	97.8%	96.1%	78.8%	82.3%	92.2%
TRBA	93.5%	91.2%	91.7%	93.8%	97.2%	97.3%	83.4%	80.6%	91.1%
ViTSTR-Base	87.0%	88.2%	86.7%	96.9%	**99.4%**	95.8%	**89.7%**	**95.6%**	**92.4%**
Fusion HC (*top 6*)	**97.8%**	95.1%	96.7%	**98.1%**	99.0%	96.6%	90.9%	93.5%	96.0%
Fusion MV-BM (*top 8*)	**97.8%**	**97.1%**	**100.0%**	**98.1%**	**99.7%**	98.4%	92.7%	96.4%	97.5%
Fusion MV-HC (*top 8*)	**97.8%**	**97.1%**	**100.0%**	**98.1%**	**99.7%**	99.1%	92.3%	**96.5%**	**97.6%**
Fusion MVCP-BM (*top 9*)	95.7%	96.1%	**100.0%**	**98.1%**	99.6%	99.0%	**92.8%**	96.4%	97.2%
Fusion MVCP-HC (*top 9*)	**97.8%**	96.1%	**100.0%**	**98.1%**	99.6%	**99.3%**	92.5%	96.3%	97.5%

† 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 6.3: Average results obtained across the datasets by combining the output of the top *N* OCR 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)	**96.0%**	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%	**97.5%**	**97.6%**	96.1%	97.2%
Top 9 (+ GRCNN)	95.7%	97.5%	97.5%	**97.2%**	**97.5%**
Top 10 (+ R^2AM)	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-LR)	95.0%	97.0%	97.0%	95.5%	96.5%

Upon analyzing the results presented in Table 6.2, 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 OCR models (MV-HC). While each model individually obtained recognition rates below 90% for at least two datasets (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 once again as we observe that the best overall model (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, which seems to impact certain models more than others (as explained in Section 6.1.2, we exploited synthetic data 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). We maintained these datasets in our experimental setup, despite their limited number of images, precisely because they provide an opportunity to uncover or corroborate such valuable insights.

Analyzing results from individual datasets reveals that combining the outputs of multiple models does not necessarily lead to significantly improved performance compared to the best model within 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 (Polikar, 2012). Illustrated in Figure 6.3 are predictions made by multiple models and the MV-HC fusion strategy for 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.

Figure 6.3: 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.

Returning to Table 6.3, we note that the majority vote-based strategies produced similar results, with the sequence-level approach (MV) performing marginally better for a given number of combined models. Our analysis suggests 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 (see Figure 6.3).

Building on Chapter 5's emphasis on the value of cross-dataset evaluation, Table 6.4 presents the results obtained on four independent datasets[18]. These particular datasets are commonly employed for such evaluations (Zou et al., 2020; Fan and Zhao, 2022; Ke et al., 2023).

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. This performance gap was most evident on the CD-HARD dataset, known for its challenges due to the predominance of heavily tilted LPs

[18] To train the models, we excluded the few images from the ChineseLP dataset that are also found in CLPD (this occurs because both collections include internet-sourced images). A thorough examination of the presence of *near-duplicates* within public datasets and its consequential impact will be carried out in Chapter 8.

Table 6.4: 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. The number of LPs in each dataset is listed below its name.

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

and the wide variety of LP layouts (as shown in Figure 6.1). Interestingly, in this cross-dataset scenario, the MV-BM strategy exhibited slightly superior performance compared to MV-HC. Unexpectedly, 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 6.5 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 6.5: 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 **accuracy**)	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^2AM)	97.2%	15.9	63	84.2	12
Top 11 (+ CRNN)	97.0%	2.9	343	87.1	11
Top 12 (+ Multi-Task-LR)	97.0%	2.3	427	89.4	11

Models (ranked by **speed**)	MV-HC	Individual		Fusion	
		Time	FPS	Time	FPS
Top 1 (Multi-Task-LR)	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^2AM)	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 models 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 6.5, combining 4-6 fast models appears to be the optimal choice for striking a better balance between speed and accuracy.

6.3 Final Remarks

This chapter examined the potential improvements in LPR results by fusing the outputs from multiple OCR models. Distinguishing itself from prior studies, our research explored 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 a majority vote at the 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 conclude 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 the recognition results while maintaining the system's efficiency at an acceptable level.

7. ADVANCING MULTINATIONAL LICENSE PLATE RECOGNITION THROUGH SYNTHETIC AND REAL DATA FUSION: A COMPREHENSIVE EVALUATION

Despite the considerable progress in the state of the art, 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 toward specific regional identifiers (Zhang et al., 2021c; Liu et al., 2021; Wang et al., 2022b; Shvai et al., 2023).

One way to mitigate this problem would be to embrace the "wildness" of the internet to collect a large-scale dataset from multiple sources (Torralba and Efros, 2011). However, labeling such a dataset would be very expensive and time-consuming (Björklund et al., 2019; Han et al., 2020; Gao et al., 2023), not to mention the growing concerns surrounding privacy (Chan et al., 2020; Kong et al., 2021; Trinh et al., 2023). In this scenario, synthetic data emerges as a practical alternative, offering a cost-effective and privacy-preserving solution while providing the diversity and scale needed for effectively training deep learning-based models.

Although recent research has explored creating synthetic LP images to improve LPR performance, our analysis in Section 7.1 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 through the integration of data generated from various methodologies. Additionally, most works have focused on LPs from a single region. To illustrate, researchers have trained separated instances of Generative Adversarial Networks (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. 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 chapter makes the following contributions:

- The most extensive experimental evaluation ever 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 (around 2k). This success stems from our approach of supplementing these real images with many synthetic ones created through character permutation 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 that surpass those reached by state-of-the-art methods and established commercial systems. These findings are particularly impressive because our models were 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;

This chapter is structured as follows. Section 7.1 outlines the prevalent methods for synthesizing LP images in the literature. Section 7.2 elaborates on our methodology for generating synthetic data, which will be integrated with real data to train the OCR models. Section 7.3 describes the experimental setup, including the datasets and models explored. The results are presented and analyzed in Section 7.4. Finally, Section 7.5 summarizes our findings.

7.1 Related Work

Many methods have been proposed to generate synthetic LP images. These methods aim to mitigate bias in the experiments and reduce the reliance on large volumes of real images for training OCR models. The subsequent paragraphs provide a concise overview of three popular methods used for this purpose.

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. Subsequently, 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 (Björklund et al., 2019; Maier et al., 2022; Gao et al., 2023). Regarding the process of creating LP images, these works primarily differed in the LP layout synthesized and the specific transformations applied. For instance, Björklund et al. (2019) focused on creating Italian LPs, Maier et al. (2022) generated German LPs, and Gao et al. (2023) synthesized LPs from mainland China. In general, the transformations applied include modifications in font thickness, pixel shifts in character positions, LP rotation, and adjustments in brightness and contrast.

Rendering-based methods face a significant limitation as they generate images with inconsistent distributions compared to real-world images, even when incorporating many transformations (Wu et al., 2019; Maier et al., 2022; Gao et al., 2023). 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.

Generating synthetic data through character permutation is a simple yet effective method for achieving balance among character classes. Essentially, 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 characters that are overrepresented in the training set with those that are underrepresented. To our knowledge, this

permutation-based approach was first explored in the LPR context by Gonçalves et al. (2018). Since then, several authors have successfully applied it to construct well-balanced training sets in terms of character classes. The following paragraph presents three examples, accompanied by a brief description of the subtle variations in how the respective authors implemented this method.

Shashirangana et al. (2022) 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. (2022) 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 researchers 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. (2021b) performed same-category permutations only (letters were swapped with other letters, and digits with other 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. (2017) with the proposal of the widely recognized pix2pix model. As detailed in Section 2.2.2.2, 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 pixelwise or patchwise labeling) can be time-consuming and even unrealistic. To tackle this challenge, subsequent works provided a novel perspective in which the proposed models (e.g., CycleGAN, DualGAN and DiscoGAN) 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. Having examined various studies employing GANs to generate synthetic data for improved LPR in Section 3.3, we will now revisit a selection of these publications relevant to this chapter's context.

Wang et al. (2022b) employed CycleGAN (Zhu et al., 2017b) to transform a large number of script LP images, created using OpenCV, into realistic ones (specific details were not provided). Similarly, Zhang et al. (2021c) 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 (2022) adopted essentially the same approach but trained CycleGAN with the Wasserstein distance loss. Their experiments focused on two distinct LP styles, one from mainland China and another from the Taiwan region.

Han et al. (2020) trained CycleGAN, StarGAN and pix2pix to generate images of the major 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. (2022) 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. (2023) 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 detailed in Section 3.3, 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. (2022b) assessed the efficacy of their strategy solely based on the results achieved by their CNN-based model. Similarly, Zhang et al. (2021c) considered only the results reached by an OCR based on Xception, and Fan and Zhao (2022) considered only the results yielded by CNNG, their multi-task recognition model. We posit that such an evaluation is suboptimal because images created through a specific method may disproportionately benefit certain approaches over others, hindering a fair evaluation of the data generation technique itself. As mentioned in Section 3.5, this phenomenon was evidenced in (Laroca et al., 2019), where two segmentation-free approaches (Multi-Task and CRNN) had a much higher performance gain than the YOLO-based CR-NET model (Silva and Jung, 2020) 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 caught our attention is that most works are still focused on LPs from a single region, even though this limitation has been acknowledged for many years in the literature (Mecocci and Tommaso, 2006; Anagnostopoulos et al., 2008). 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 (Han et al., 2020; Maier et al., 2022; Shvai et al., 2023). Researchers often opted to train separate instances of the proposed models for each layout. For example, one model generates/recognizes LPs from the Taiwan region, another model generates/recognizes LPs from mainland China, and so forth (Björklund et al., 2019; Zhang et al., 2021d; Wang et al., 2022c). However, 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 need to adjust parameters and retrain models when incorporating support for LPs from new regions or even markedly 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.

7.2 Synthetic Data

This section details our approach for generating synthetic data, which will be combined with real data to train the deep models for LPR. We first describe the methodology adopted 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.

7.2.1 Templates

While there are various approaches for creating LP images using templates, the method employed in this chapter is quite straightforward. First, blank templates that match the aspect ratio and color scheme of real LPs are sourced from the internet[19]. Subsequently, a sequence of characters, selected randomly yet crafted to mirror the patterns found on authentic LPs, is superimposed onto each template. Figure 7.1 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.

Figure 7.1: Examples of the template-based LP images we created for this study. Notably, any sequence can be generated for each template. The background and character images were manually gathered from the internet[19]. During training, these LP images are subjected to various transformations to introduce variability.

To better simulate real-world scenarios, the templates we generated using this method were derived from the LP styles observed within the training sets of the datasets explored in our experiments (refer to Section 7.3.2 for details). In other words, we did not create templates for LP styles found exclusively in the test sets. To illustrate, one of the datasets we employed in our cross-dataset assessments contains images of electric vehicles registered in mainland China, which feature 8-character green LPs. Despite this, we refrained 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 7.1, images produced by such rendering-based approaches often exhibit inconsistent distributions compared to real-world images (even with transformations applied). Second, sourcing background and character images online for certain LP styles, particularly those less popular or recently introduced, can pose a challenge. This challenge played a 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 generated 100k LP images employing this approach, a number determined through preliminary experiments that showed slightly improved outcomes compared to using 50k images and similar performance to using 200k images. The number of synthesized LPs was balanced across the six explored LP layouts (i.e., American, Brazilian, Chinese, European, Mercosur, and Taiwanese), and the LP sequences were defined to maximize class balance for each character position.

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

7.2.2 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 7.1), we first expanded the bounding boxes of smaller characters, incorporating small portions of the LP background into them, so that all characters have identical dimensions. Subsequently, we replaced patches of characters that were overrepresented in the training set with patches from those that were underrepresented. To maintain consistency in illumination, we limited character permutation to patches within the same LP.

In Figure 7.2, 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 (Björklund et al., 2019; Wang et al., 2022c; Liu et al., 2024b). 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. (2018) 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 7.2 as well).

Figure 7.2: 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.

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.

7.2.3 Image-To-Image Translation (pix2pix)

As outlined in Section 7.1, 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 exploited 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. (2021b) provided labels for most of these images, we further enriched the annotations by labeling the positions of the LP corners.

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 (Isola et al., 2017) 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 frameworks such as Chainer, Keras, PyTorch, TensorFlow, Torch, and others[20]. This choice was 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 was prepared as follows. For each LP image generated through character permutation, which serves as the intended output, a corresponding segmentation mask was created to serve as the input. These masks were designed such that each color represents a distinct LP layout class or character class. For example, as shown in Figure 7.3, 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[21] was employed to generate a set of colors that were maximally distinguishable from each other. Black (0, 0, 0) and similar shades were 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.

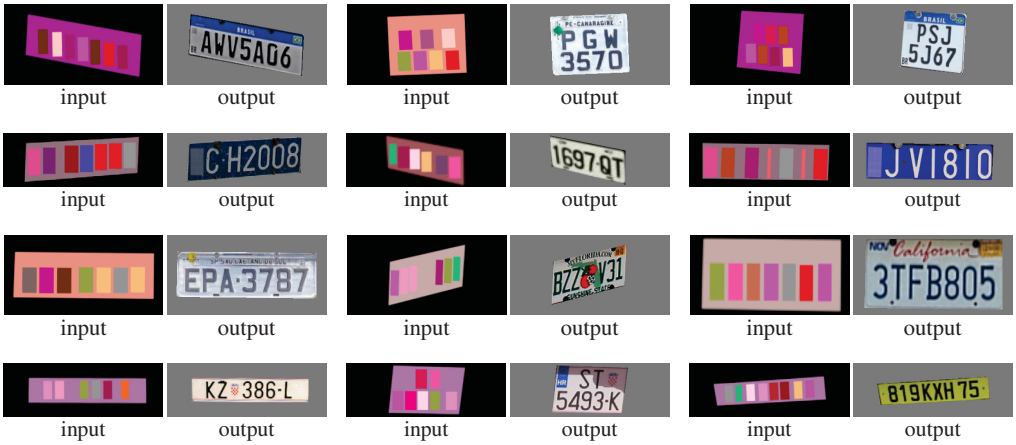


Figure 7.3: Examples of image pairs used for training the pix2pix model. To create the input masks, labels are required for both the LP's layout and corners, as well as for the bounding box of each character.

After completing the model's training, the next step involves using it to generate hundreds of thousands of new LP images. Intuitively, this task was accomplished by feeding the model with segmentation masks derived from randomly selected LP layouts and character sequences. While the characters were sampled from the valid alphabet per position, we ensured 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

[20] 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`.

[21] The Glasbey library is available at `https://github.com/taketwo/glasbey`

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 Chapter 6, 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 LP layouts, totaling 300k images. This strategy proved effective in filtering out most images with defects, although it may have led to the exclusion of some instances with a higher degree of variability. Examples of the selected images are shown in Figure 7.4.

Figure 7.4: Examples of selected images from those generated using pix2pix. From top to bottom, we show American, Brazilian, Chinese, European, Mercosur, and Taiwanese LPs.

It should be noted that we trained the pix2pix model to produce a blurred representation instead of Chinese characters (this can be seen in Figures 7.3 and 7.4). This adjustment was 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. Further details on how we handled Chinese characters in our experiments can be found in Section 7.3.3.

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 (Zhang et al., 2021c; Fan and Zhao, 2022; Wang et al., 2022b).

The major limitation of this GAN-based method stems from its reliance on the training data, as it cannot synthesize LP layouts that are not included in the training set (Gao et al., 2023).

7.3 Experimental Setup

This section describes the experimental setup adopted in this chapter. We begin by outlining the OCR models implemented for our evaluations. Subsequently, we list the datasets employed, which are the same used in the previous chapter, while briefly reminding the reader of their

characteristics such as the number of images, resolution, and LP layouts. Lastly, we elaborate on the methodology used for performance evaluation.

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 2,133 MHz, an NVMe SSD with read and write speeds of 3,500 MB/s and 3,000 MB/s respectively, and an NVIDIA Quadro RTX 8000 GPU (48 GB).

7.3.1 OCR Models

This chapter expands upon the 12 models explored in Chapters 5 and 6 by integrating four additional models into the experiments: Table 7.1 presents an overview of all 16 models, including their original applications and the frameworks used for their implementation. We are unaware of any work in ALPR research where so many OCR models were explored.

Table 7.1: The 16 OCR models explored in this chapter.

Model	Original Application
Framework: PyTorch (Atienza, 2022)	
R^2AM (Lee and Osindero, 2016)	Scene Text Recognition
RARE (Shi et al., 2016)	Scene Text Recognition
STAR-Net (Liu et al., 2016)	Scene Text Recognition
CRNN (Shi et al., 2017)	Scene Text Recognition
GRCNN (Wang and Hu, 2017)	Scene Text Recognition
Rosetta (Borisyuk et al., 2018)	Scene Text Recognition
TRBA (Baek et al., 2019)	Scene Text Recognition
ViTSTR-Base (Atienza, 2021b)	Scene Text Recognition
ViTSTR-Small (Atienza, 2021b)	Scene Text Recognition
ViTSTR-Tiny (Atienza, 2021b)	Scene Text Recognition
Framework: Keras (Chollet et al., 2024)	
Holistic-CNN (Špaňhel et al., 2017)	License Plate Recognition
Multi-Task (Gonçalves et al., 2018)	License Plate Recognition
Multi-Task-LR (Gonçalves et al., 2019)	License Plate Recognition
CNNG (Fan and Zhao, 2022)	License Plate Recognition
Framework: Darknet (Bochkovskiy, 2023)	
CR-NET (Silva and Jung, 2020)	License Plate Recognition
Fast-OCR (Laroca et al., 2021a)	Image-based Meter Reading

As in previous chapters, the YOLO-based models (i.e., CR-NET and Fast-OCR) were implemented using Darknet (Bochkovskiy, 2023); the multi-task models (those listed in the middle section of Table 7.1) were implemented using Keras (Chollet et al., 2024); and the other models were implemented using a popular fork of the open source repository of Clova AI Deep Text Recognition Benchmark (Atienza, 2022). For training the models within each framework, we used the same hyperparameters as in previous chapters (refer to Section 5.1).

7.3.2 Datasets

We used the same 12 datasets explored in the previous chapter, as shown in Table 7.2. We also adhered to the same data-splitting protocol established earlier. This means that eight datasets were used to train, validate and test the chosen models (intra-dataset experiments), while the remaining four datasets were used solely for testing their generalizability (cross-dataset experiments). For detailed information on how each dataset was divided in the intra-dataset setup, see Section 6.1.2.

In line with the experiments conducted in the preceding chapters, we employed Albumentations (Buslaev et al., 2020) to balance the number of training images from different datasets. This

Table 7.2: The 12 datasets used in the experiments carried out for this chapter.

Dataset	Images	Resolution	LP Layout
Caltech Cars (Weber, 1999)	126	896 × 592	American
EnglishLP (Srebrić, 2003)	509	640 × 480	European
UCSD-Stills (Dlagnekov and Belongie, 2005)	291	640 × 480	American
ChineseLP (Zhou et al., 2012)	411	Various	Chinese
AOLP (Hsu et al., 2013)	2,049	Various	Taiwanese
OpenALPR-EU* (OpenALPR, 2016)	108	Various	European
SSIG-SegPlate (Gonçalves et al., 2016a)	2,000	1920 × 1080	Brazilian
PKU* (Yuan et al., 2017)	2,253	1082 × 727	Chinese
UFPR-ALPR (Laroca et al., 2018)	4,500	1920 × 1080	Brazilian
CD-HARD* (Silva and Jung, 2018)	102	Various	Various
CLPD* (Zhang et al., 2021c)	1,200	Various	Chinese
RodoSol-ALPR	20,000	1280 × 720	Brazilian & Mercosur

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

involved applying common transformations to the original images, such as random perspective shifts, random noise addition, and random adjustments to hue, saturation, and brightness.

7.3.3 Performance Evaluation

In this chapter, we detected the LPs in the original images using YOLOv4-CSP (Wang et al., 2021a) and rectified them through a combination of CDCC-NET (Laroca et al., 2021a) – for locating the LP corners – and perspective transformation (the rectification process is detailed in the next paragraph). These models were chosen due to their remarkable performance in balancing the trade-off between robustness and efficiency in the studies they were proposed. We adopted this procedure to fairly compare our results with end-to-end ALPR systems and to better simulate real-world scenarios, where the LPs are not always optimally detected.

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 7.5. Recent works that exploited LP rectification to improve the recognition results include (Qin and Liu, 2022; Xu et al., 2022; Jiang et al., 2023b).

(a) detected LP regions (b) rectified LP regions

Figure 7.5: 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.

It is essential to highlight that 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 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.

In this chapter, we also adhere to the methodology established by Li et al. (2019), 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 7.4.2.2) were obtained in the same way, disregarding Chinese characters.

7.4 Results and Discussion

This section presents and analyzes the outcomes of our experiments. Section 7.4.1 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 7.4.2 then delves into a detailed examination of the end-to-end results obtained by employing different OCR models.

7.4.1 LP Detection and Corner Detection

To evaluate detection tasks, one can employ various quantitative criteria. Our assessment includes the widely adopted precision, recall and f-score metrics (described in Section 2.1). In line with recent studies (Jiang et al., 2023b; Ke et al., 2023), for this chapter, we define the detections as correct when the Intersection over Union (IoU) with the ground truth exceeds 0.7.

Table 7.3 presents the results obtained by YOLOv4-CSP (Wang et al., 2021a) and IWPOD-NET (Silva and Jung, 2022) in the LPD stage. 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 (akin to what was observed in Chapters 5 and 6); (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 (leading to lower recall rates). These observations likely influenced the decision of Silva and Jung (2022) to feed regions identified by a vehicle detector (YOLOv3) into IWPOD-NET instead of applying it directly to the original image. It is worth noting that optimizing both precision and recall is crucial for efficient system operation, as it relies on the detection of all LPs with minimal false positives.

Table 7.3: Results obtained by YOLOv4-CSP and IWPOD-NET in the LPD 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%

To rectify the LPs found by YOLOv4-CSP, it is necessary to locate the four corners associated with each of them. Table 7.4 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 (Jia and Xie, 2023), 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:

$$LP\text{-}NME(C, \hat{C}) = \frac{1}{4} \sum_{i=1}^{4} \frac{||C_i - \hat{C}_i||}{d} , \tag{7.1}$$

where C and $\hat{C}$ are the ground truth and predicted corners, respectively, and d is the normalization factor. Following Jia and Xie (2023), we adopt the diagonal length of the smallest bounding box that completely encloses the LP as the normalization factor.

Table 7.4: 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.

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
LocateNet (Meng et al., 2018)	0.0739	0.0359	0.0782	0.1092	0.0730	0.0329	0.0556	0.0592	0.0647
Hybrid-MobileNetV2 (Yoo and Jun, 2021)	0.0323	0.0226	0.0352	0.0391	0.0332	0.0214	0.0313	0.0383	0.0317
IWPOD-NET (Silva and Jung, 2022)	0.0244	0.0143	0.0205	0.0138	0.0205	0.0098	0.0194	0.0141	0.0171
CDCC-NET (Laroca et al., 2021a)	0.0160	0.0117	0.0164	0.0176	0.0142	0.0098	0.0168	0.0150	**0.0147**

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 7.6 showcases the predictions made by all models for five LP images. While some predictions show clear similarities across models, the CDCC-NET model exhibits superior overall accuracy.

Figure 7.6: Qualitative results achieved by four different models in corner detection. For better viewing, we draw a polygon from the predicted corner positions.

The findings outlined in this section substantiate our choice to employ YOLOv4-CSP for LPD and CDCC-NET for corner detection. As elaborated in Section 7.3.3, the corners predicted by CDCC-NET are used to rectify the LPs before recognition.

7.4.2 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 model with those reached by state-of-the-art approaches and established commercial systems (Sections 7.4.2.1 to 7.4.2.3). 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 7.4.2.4 examines the trade-off between speed and accuracy exhibited by the recognition models, highlighting those that strike a favorable balance.

7.4.2.1 Intra-Dataset Experiments

Table 7.5 presents the end-to-end results obtained across the disjoint test sets of the eight datasets used to train and validate 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 7.2. Later in this section, we present an ablation study that details the contribution of each image synthesis method to the results achieved. Importantly, Table 7.5 also includes the outcomes achieved by combining the outputs of different models, following the optimal strategy identified in the previous chapter. The results demonstrate that combining synthetic data and model fusion further enhances LPR performance.

Table 7.5: 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 (Fan and Zhao, 2022)	**97.8%**	91.2%	96.7%	98.8%	99.1%	98.8%	**96.1%**	97.1%	96.9%
CR-NET (Silva and Jung, 2020)	93.5%	96.1%	98.3%	96.9%	98.7%	98.0%	89.3%	88.3%†	94.9%
CRNN (Shi et al., 2017)	93.5%	96.1%	96.7%	95.7%	98.8%	97.5%	87.0%	92.2%	94.7%
Fast-OCR (Laroca et al., 2021a)	95.7%	97.1%	95.0%	96.9%	98.7%	96.0%	89.6%	88.1%†	94.6%
GRCNN (Wang and Hu, 2017)	**97.8%**	**99.0%**	96.7%	98.8%	99.0%	97.9%	87.4%	93.0%	96.2%
Holistic-CNN (Špaňhel et al., 2017)	95.7%	91.2%	93.3%	99.4%	99.3%	98.4%	94.9%	**97.9%**	96.3%
Multi-Task (Gonçalves et al., 2018)	**97.8%**	94.1%	**100.0%**	98.8%	99.1%	98.6%	93.3%	95.1%	97.1%
Multi-Task-LR (Gonçalves et al., 2019)	95.7%	93.1%	93.3%	**100.0%**	99.6%	97.5%	94.6%	96.6%	96.3%
R^2AM (Lee and Osindero, 2016)	**97.8%**	94.1%	95.0%	98.8%	99.3%	99.3%	90.6%	94.4%	96.1%
RARE (Shi et al., 2016)	**97.8%**	97.1%	98.3%	98.1%	99.4%	99.1%	91.9%	96.5%	97.3%
Rosetta (Borisyuk et al., 2018)	95.7%	98.0%	98.3%	98.1%	98.7%	98.3%	92.6%	96.0%	97.0%
STAR-Net (Liu et al., 2016)	**97.8%**	**99.0%**	98.3%	98.1%	99.1%	99.3%	94.7%	97.0%	**97.9%**
TRBA (Baek et al., 2019)	**97.8%**	**99.0%**	98.3%	98.8%	98.8%	99.3%	94.0%	97.3%	**97.9%**
ViTSTR-Base (Atienza, 2021b)	95.7%	96.1%	93.3%	99.4%	**99.9%**	**99.4%**	94.6%	97.7%	97.0%
ViTSTR-Small (Atienza, 2021b)	95.7%	96.1%	98.3%	98.1%	99.1%	98.5%	94.9%	96.8%	97.2%
ViTSTR-Tiny (Atienza, 2021b)	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%
Model Fusion MV-HC (top 8)	97.8%	99.0%	100.0%	99.4%	99.4%	100.0%	98.2%	98.6%	99.1%

†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.

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 (Zhang et al., 2021a; Zhou et al., 2023; Ding et al., 2024). 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 7.7 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.

Figure 7.7: 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 7.3.3.

A compelling aspect to consider is the impact of synthetic data in scenarios with limited availability of training data, as public datasets collected in certain regions often have a restricted number of images. Table 7.6 presents the average recognition rates attained by STAR-Net and TRBA when trained with reduced portions – 50%, 25%, 10%, 5% and 1% – of the original training data, with and without the addition of synthetic data. 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 7.6: 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 7.7 elucidates the effectiveness of each image synthesis method described in Section 7.2, as well as their combination, to the results obtained. It reveals that each method contributes considerably to enhancing the results. Notably, a substantial synergistic effect is observed when combining these methods, pushing the performance boundaries of OCR 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. When real images were combined solely with

images generated through character permutation, as in (Laroca et al., 2021b; Shashirangana et al., 2022), 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 (Maier et al., 2022; Gao et al., 2023), 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 (Zhang et al., 2021c; Shvai et al., 2023), yielded average recognition rates of 93.2% and 95.2% for unrectified and rectified LPs, respectively.

Table 7.7: Average recognition rates obtained across all models and datasets with different types of images included in the training set. 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, our evaluation of their integration is limited to cases where real images coexist in the training set. 'Data aug.' refers to images created by applying common transformations.

Real Images + data aug.	Templates	Permutation	GAN (pix2pix)	Average	Average (rect.)
	✓			42.5%	46.5%
✓				84.5%	88.1%
✓		✓		91.4%	93.6%
✓	✓			92.5%	94.7%
✓			✓	93.2%	95.2%
✓	✓	✓		93.8%	95.5%
✓		✓	✓	94.0%	95.6%
✓	✓		✓	94.1%	95.8%
✓	✓	✓	✓	**94.9%**	**96.4%**

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%).

Interestingly, 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 finding aligns with the fact that images created via character permutation still share many characteristics with their original counterparts (e.g., character position, compression artifacts, and camera noise) despite having different sequences of characters.

While not the primary focus of Table 7.7, it also reinforces the importance of rectifying the LPs before the recognition stage, as this consistently resulted in improved outcomes.

7.4.2.2 Cross-Dataset Experiments

As emphasized throughout this work, conducting cross-dataset experiments is pivotal in assessing the models' generalizability. Thus, Table 7.8 presents the recognition rates obtained by all models on the four datasets not seen during the training stage: OpenALPR, PKU, CD-HARD and CLPD.

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 (Baek et al., 2019), 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. That is why we consider *YOLOv4-CSP* (detection) + *CDCC-NET* (rectification) + *TRBA* (recognition) to be our best approach and therefore employ it in the comparisons with state-of-the-art approaches in the next section.

Table 7.8: 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.

Model \ Dataset # LPs	OpenALPR-EU # 108	PKU # 2,253	CD-HARD # 104	CLPD # 1,200	Average
CNNG (Fan and Zhao, 2022)	95.4%	98.6%	58.7%	92.9%	86.4%
CR-NET (Silva and Jung, 2020)	93.5%	**99.5%**	67.3%	92.9%	88.3%
CRNN (Shi et al., 2017)	97.2%	99.1%	56.7%	94.2%	86.8%
Fast-OCR (Laroca et al., 2021a)	98.1%	99.1%	69.2%	94.4%	90.2%
GRCNN (Wang and Hu, 2017)	97.2%	99.0%	57.7%	94.5%	87.1%
Holistic-CNN (Špaňhel et al., 2017)	95.4%	99.0%	54.8%	94.0%	85.8%
Multi-Task (Gonçalves et al., 2018)	96.3%	98.8%	54.8%	93.7%	85.9%
Multi-Task-LR (Gonçalves et al., 2019)	94.4%	98.8%	53.8%	92.6%	84.9%
R^2AM (Lee and Osindero, 2016)	98.1%	99.4%	57.7%	93.8%	87.3%
RARE (Shi et al., 2016)	**99.1%**	99.1%	72.1%	95.2%	91.4%
Rosetta (Borisyuk et al., 2018)	97.2%	99.2%	64.4%	93.8%	88.7%
STAR-Net (Liu et al., 2016)	98.1%	98.5%	71.2%	95.0%	90.7%
TRBA (Baek et al., 2019)	**99.1%**	99.4%	**76.9%**	**96.2%**	**92.9%**
ViTSTR-Base (Atienza, 2021b)	94.4%	99.0%	54.8%	93.4%	85.4%
ViTSTR-Small (Atienza, 2021b)	96.3%	97.4%	59.6%	94.3%	86.9%
ViTSTR-Tiny (Atienza, 2021b)	94.4%	97.6%	53.8%	92.3%	84.5%
Average	96.5%	98.8%	61.5%	93.9%	87.7%
Model Fusion MV-HC (top 8)	99.1%	99.6%	81.7%	97.6%	94.5%

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 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.

A noteworthy insight from Table 7.8 is that integrating synthetic data with model fusion also improves LPR performance in cross-dataset scenarios.

7.4.2.3 Comparison With Previous Works and Commercial Systems

In Table 7.9, we compare the end-to-end results achieved by our best approach with those reported by state-of-the-art ALPR systems. Following common practice, to ensure fairness, we only consider systems evaluated in the same way as in our benchmark (see details in Section 7.3.2). We also compare our results with those obtained by the Sighthound (2023) and OpenALPR (2023) commercial systems (details on these systems were provided in Section 3.4).

It is impressive that, without using any heuristics rule or post-processing, our best approach (TRBA) achieves state-of-the-art performance on all datasets except AOLP. Note that we actually attained state-of-the-art results (e.g., 99.9%) in this dataset when employing other models for LPR (see Table 7.5); however, we do not consider those results here as the respective models did not perform better than TRBA on average.

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 Sighthound (2023) and OpenALPR (2023) commercial systems performed poorly on the RodoSol-ALPR dataset (with 57.0% and 69.3% recognition rates, respectively). As previously discussed in Chapter 5 and now detailed in Table 7.10, the primary reason for such underwhelming results is the limited effectiveness of these systems in handling motorcycle LPs (which have two-row character arrangement and smaller size) and Mercosur LPs. These

Table 7.9: 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 (intra-dataset experiments). The best results achieved in each dataset are shown in bold.

Approach \ Test set	Caltech Cars # 46	EnglishLP # 102	UCSD-Stills # 60	ChineseLP # 161	AOLP # 687	SSIG-SegPlate # 804	UFPR-ALPR # 1,800	RodoSol-ALPR # 8,000	Average
Sighthound (2023)	87.0%	93.1%	96.7%	95.0%	95.5%	82.8%	62.9%	57.0%	83.7%
Castro-Zunti et al. (2020)[‡]	91.3%	–	**98.3%**	–	–	–	–	–	–
Silva and Jung (2022)	–	–	–	–	97.4%	–	86.3%	–	–
Henry et al. (2020)	**97.8%**	97.1%	–	–	98.9%	–	–	–	–
Laroca et al. (2021b) (run 1)[†]	**97.8%**	96.1%	96.7%	98.1%	**99.4%**	98.8%	89.7%	–	–
Zhou et al. (2023)	–	–	–	–	–	–	90.3%	–	–
Silva and Jung (2022)[†]	–	–	–	–	99.0%	–	91.8%	–	–
OpenALPR (2023)[†]	95.7%	98.0%	**98.3%**	96.9%	97.1%	93.0%	92.2%	69.3%	92.6%
Chen et al. (2023)	–	–	–	–	–	–	–	96.6%	–
Nascimento et al. (2023)[‡]	–	–	–	–	–	–	–	96.6%	–
Ours	87.0%	91.2%	88.3%	98.1%	98.4%	98.1%	92.1%	96.8%	93.7%
Zhang et al. (2021a)	–	–	–	–	–	98.6%	92.3%	–	–
Liu et al. (2024a)[‡]	–	–	–	–	99.0%	–	–	97.0%	–
Ours + synthetic	**97.8%**	**99.0%**	**98.3%**	**98.8%**	98.8%	**99.3%**	**94.0%**	**97.3%**	**97.9%**

[†] ALPR systems that rely on pre-defined heuristic rules (prior knowledge) to refine the predictions returned by the OCR model.

[‡] The LP patches fed into the OCR model were cropped directly from the ground truth in (Castro-Zunti et al., 2020; Nascimento et al., 2023; Liu et al., 2024a).

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).

Table 7.10: 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.

System	Vehicle Type		LP Layout	
	Cars	Motorcycles	Brazilian	Mercosur
Sighthound (2023)	81.3%	32.7%	63.9%	50.1%
OpenALPR (2023)	95.6%	43.0%	90.7%	47.8%

There are many recent works where the authors evaluated the generalizability of the proposed methods in the PKU (Yuan et al., 2017) and CLPD (Zhang et al., 2021c) datasets, both collected in mainland China. Hence, in Table 7.11, we compare the results obtained by these methods (plus Sighthound and OpenALPR) with those reached by our best approach. For each method, we also provide details on the number of real Chinese LPs used for its training, as well as its multinational applicability (we classify methods as multinational if they were not trained or fine-tuned exclusively on Chinese LPs).

When exploring synthetic data for training the OCR model, our 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 (Xu et al., 2018). Indeed, this is one of the reasons why our approach 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). By incorporating LP images extracted from CCPD's training set into our training data, mirroring previous studies, our approach achieved impressive recognition rates of 97.3% and 99.5% on the CLPD and PKU datasets, respectively.

7.4.2.4 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 (Jiang et al., 2023b; Ke et al., 2023; Ding et al.,

Table 7.11: Comparison of the recognition rates obtained by our best approach (which uses TRBA as the recognition model), 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.

Approach	Real images of Chinese LPs used for training	Multinational	Recognition Rate CLPD	Recognition Rate PKU
Sighthound (2023)	?	✓	85.2%	89.3%
Zhang et al. (2021c)	100,000+		87.6%	90.5%
Fan and Zhao (2022)	100,000+	✓	88.5%	92.5%
Ours	506	✓	90.1%	96.8%
Rao et al. (2024)†	4,444		91.4%	96.1%
Liu et al. (2021)	10,000		91.7%	–
OpenALPR (2023)	?		91.8%	96.0%
Chen et al. (2023)	100,000+		92.4%	92.8%
Ke et al. (2023)	100,000+		93.2%	–
Zou et al. (2020)	100,000+		94.0%	96.6%
Zou et al. (2022)	100,000+		94.5%	–
Wang et al. (2022b)	100,000+		94.8%	–
Wang et al. (2022c)	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%**

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

2024). Thus, this section examines the speed/accuracy trade-off of the OCR models explored in this chapter. Figure 7.8 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.

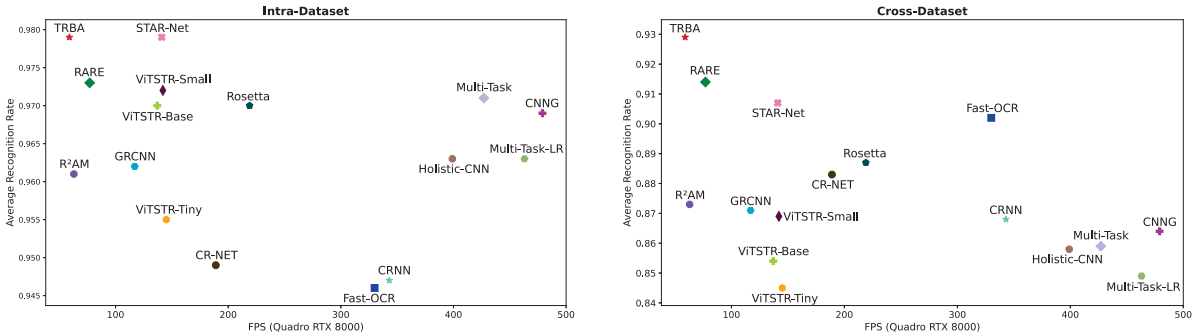


Figure 7.8: Average recognition rate across datasets and the corresponding FPS processing capabilities for all OCR models on intra-dataset (left) and cross-dataset (right) experiments. The specific FPS value for each model is as follows: CNNG: 479; CR-NET: 189; CRNN: 343; Fast-OCR: 330; GRCNN: 117; Holistic-CNN: 399; Multi-Task: 427; Multi-Task-LR: 463; R^2AM: 63; RARE: 77; Rosetta: 219; STAR-Net: 141; TRBA: 59; ViTSTR-Base: 137; ViTSTR-Small: 142; and ViTSTR-Tiny: 145.

In *intra-dataset* scenarios, the multi-task models, particularly Multi-Task and CNNG, demonstrated an exceptional balance between speed and accuracy. This success stems from their ability to learn potential classes for each character position independently, avoiding confusion between similar letters and digits in layouts where they appear in distinct positions. If the primary goal is to achieve the utmost recognition rate across various scenarios, STAR-Net stands as a more compelling option compared to TRBA. This is because STAR-Net reached 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 7.4.2.2, 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 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 (Atienza, 2021b).

7.5 Final Remarks

This chapter 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 emerged from our study. Primarily, the massive use of synthetic data significantly improved the performance of all models. Both 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 either 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 in both intra- and cross-dataset scenarios. 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. 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 provided three important insights. First, each synthesis method contributed considerably to enhancing the results, and a substantial synergistic effect was observed when combining them. This finding contrasts with the common practice of generating synthetic LPs exclusively through a single methodology. Second, incorporating synthetic data into the training set enabled 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. Third, consistent with findings from prior research, rectifying the LPs before the recognition stage proved essential for achieving optimal LPR performance.

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.

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 OCR models under investigation (refer to Table 7.7), 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 LPD and corner detection tasks, as reported in Tables 7.3 and 7.4. As mentioned earlier in this work, a single training process for some models (e.g., TRBA and ViTSTR-Base) takes several days to complete on an NVIDIA Quadro RTX 8000 GPU, which is currently one of the top-performing GPUs in the market.

8. DO WE TRAIN ON TEST DATA? THE IMPACT OF NEAR-DUPLICATES ON LICENSE PLATE RECOGNITION

LPR methods are typically evaluated using images from public datasets, which are divided into disjoint training and test sets using standard splits, defined by the datasets' authors, or following previous works (when there is no standard split). In many cases, such an assessment is carried out independently for each dataset (Laroca et al., 2018; Zhuang et al., 2018; Weihong and Jiaoyang, 2020; Zhang et al., 2021d; Ke et al., 2023; Pham, 2023).

Although the images for training and testing belong to disjoint sets, the splits traditionally adopted in the literature were defined without considering that the same LP may appear in multiple images. As a result, we found that there are many *near-duplicates* (i.e., different images of the same LP) in the training and test sets of datasets widely explored in ALPR research (see Section 8.1.1). In this chapter, to evaluate the impact of such duplicates on LPR, we focus our analysis on the AOLP (Hsu et al., 2013) and CCPD (Xu et al., 2018) datasets, as they are the most popular datasets in the field. Nevertheless, Section 8.3 highlights the existence of near-duplicates in several other datasets and gives examples of how it has been overlooked in the literature.

Considering that recent ALPR approaches rectify (unwarp) the detected LPs before feeding them to the recognition model (Fan and Zhao, 2022; Qin and Liu, 2022; Silva and Jung, 2022; Wang et al., 2022c; Xu et al., 2022; Jiang et al., 2023b), the presence of duplicates in the training and test sets means that LPR models are, in many cases, being trained and tested on essentially the same images (see Figure 8.1). This is a critical issue for accurate scientific evaluation (Barz and Denzler, 2020; Emami et al., 2020). Researchers aim to compare models in terms of their ability to generalize to unseen data (Feldman and Zhang, 2020; Liao et al., 2021). With a considerable number of duplicates, however, there is a risk of comparing the models in terms of their ability to memorize training data, which increases with the model's capacity (Barz and Denzler, 2020; Hooker et al., 2020).

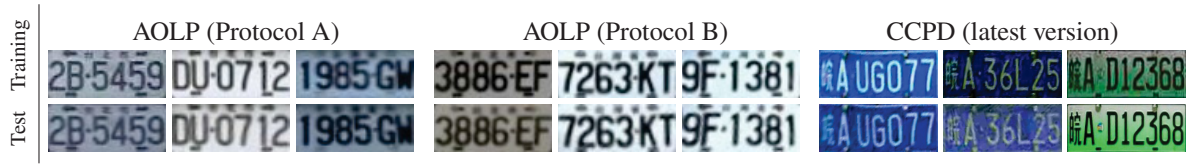


Figure 8.1: Examples of near-duplicates in the training and test sets of the AOLP and CCPD datasets, which are by far the two most popular datasets in the LPR literature. The top row shows LPs cropped and rectified from images in the training sets, while the bottom row shows LPs cropped and rectified from their nearest neighbors in the respective test set. We show three image pairs for each dataset representing the 10th, 50th and 90th percentiles based on their Euclidean distance in pixel space. Protocols A and B in the AOLP dataset are described in Section 8.1.1.

In light of this, we create *fair splits* for the AOLP and CCPD datasets (see Section 8.2.1) and compare the performance of six well-known OCR models applied to LPR under the original (adopted in previous works) and fair protocols[22]. Our results indicate that the presence of duplicates greatly affects the performance evaluation of these models. Considering the experiments under the AOLP-B protocol as an example, the model that reached the best results under the traditional split ranked third under the fair one. Such results imply that the duplicates have biased the evaluation and development of deep learning-based models for LPR.

[22] An article version of this chapter was accepted for presentation at the *2023 International Joint Conference on Neural Networks (IJCNN)* (Laroca et al., 2023a).

This chapter builds upon the work of Barz and Denzler (2020), who identified duplicates within the CIFAR-10 and CIFAR-100 datasets. It is further motivated by the results presented in Chapter 5, particularly the substantial drops observed in LPR performance when training and testing state-of-the-art models in a leave-one-dataset-out experimental setup.

In summary, this chapter has two main contributions:

- We reveal the presence of near-duplicates in the training and test sets of datasets widely adopted in the ALPR literature. Our analysis shows the impact of such duplicates on the evaluation of six well-known recognition models applied to LPR.
  - Our results on the AOLP dataset indicate that the high fraction of near-duplicates in the splits traditionally employed in the literature may have hindered the development and acceptance of more efficient LPR models that have strong generalization abilities but do not memorize duplicates as well as other models;
  - Our experiments on the CCPD dataset give a clearer picture of the true capabilities of LPR models compared to prior evaluations using the standard split, in which the test set has duplicates in the training set. Results revealed a decrease in the average recognition rate from 80.3% to 77.6% when the experiments were conducted under a fair split without duplicates.
- We create and release *fair splits* for these datasets where there are no duplicates in the training and test sets, and the key characteristics of the original partitions are preserved as much as possible (see details on Section 8.2.1).

This chapter is structured as follows. We describe the AOLP and CCPD datasets in Section 8.1, detailing the protocols often adopted for each and how many near-duplicates they have. Section 8.2 details the experiments performed. The presence of duplicates in other popular datasets is discussed in Section 8.3. Finally, conclusions are provided in Section 8.4.

8.1 The AOLP and CCPD Datasets

The two most popular datasets for ALPR (in terms of the number of works that explored them) are AOLP (Hsu et al., 2013) and CCPD (Xu et al., 2018). While most authors explored at least one of these two datasets in their experiments (Li et al., 2019; Silva and Jung, 2022; Dai et al., 2024), there are many works in which the experiments were performed exclusively on them (Xie et al., 2018; Zhang et al., 2020c; Liang et al., 2022; Pham, 2023).

AOLP was created to verify that ALPR is better handled in an application-oriented way. It is categorized into three subsets: access control (AC), traffic law enforcement (LE), and road patrol (RP). These subsets have 681, 757 and 611 images, respectively, all captured in the Taiwan region.

The AOLP dataset lacks a standardized division for training and testing purposes, leading researchers to adopt various approaches. For instance, some authors (e.g., Xie et al. (2018); Laroca et al. (2021b); Liang et al. (2022)) randomly divided its images into training and test sets with a 2:1 ratio (we refer to this protocol as *AOLP-A*). Others, including Li et al. (2019); Zhang et al. (2021d); Wang et al. (2022c), used images from different subsets for training and testing. For example, Fan and Zhao (2022); Nguyen (2022); Qin and Liu (2022) used images from the AC and LE subsets to train the proposed models and tested them on the RP subset (we refer to this protocol as *AOLP-B*). Zhuang et al. (2018) evaluated their method under both the AOLP-A and AOLP-B protocols. As commonly done in previous works, we consider that 20% of the training images are allocated for validation in both protocols.

Xu et al. (2018) claimed that the ALPR datasets available at the time (including AOLP) either lacked quantity (i.e., they had less than 10k images) or diversity (i.e., they were collected by static cameras or in overly controlled settings). Thus, to assist in better benchmarking ALPR approaches, they presented the CCPD dataset.

CCPD comprises images taken with handheld cameras by workers of a roadside parking management company on the streets of a capital city in mainland China. The dataset was updated/expanded twice after being introduced in 2018[23]. It originally consisted of 250k images, divided into subsets (e.g., Blur, Challenge, Rotate, Weather, among others) according to their characteristics (Xu et al., 2018). Then, in 2019, the authors released a new version – much more challenging than the previous one – containing over 300k images, refined annotations, and a standard split. In summary, in this protocol, the 200k images in the "Base" subset are split into training and validation sets (50%/50%), while all images from the other subsets are employed for testing. Finally, in 2020, the authors included a new subset (Green) with 11,776 images of electric vehicles, which have green LPs with eight characters (all the other subsets have images of vehicles with blue LPs containing seven characters). The Green subset has a standard split, with 49% of the images allocated for training, 8.5% for validation, and 42.5% for testing. This latest iteration of CCPD (2020) is the version explored in this chapter.

8.1.1 Duplicates

The problem with these split protocols is that they do not account for the same vehicle/LP appearing in multiple images, including images from different subsets, as shown in Figure 8.2 and Figure 8.3. While one may claim that such images have enough variety to be used both for training and testing LP detectors, as they are fed the entire images, not just the LP region, it seems reasonable to consider that such images should not be employed in the same way (i.e., for both training and testing) in the recognition stage, as the LPs look very similar after being cropped and rectified. In fact, they can look very similar even without rectification (e.g., see (d) and (e) in Figure 8.2).

In the AOLP dataset, considering the *AOLP-A* split protocol[24], there are 320 duplicates from the test set in the training one. As there are 683 test images in this protocol, **46.9%** of them have duplicates. Startlingly, the number of duplicates is even higher in the *AOLP-B* split protocol, where 413 of the 611 test images (**67.6%**) have duplicates in the training set.

The situation is less severe – albeit still concerning – for the CCPD dataset, where we found 29,943 duplicates from the test set in the training set. Despite the much higher number of duplicates in absolute terms, CCPD's current version has ≈157k images with labeled LPs in the test set; that is, the duplicates amount to **19.1%** of the test images.

8.2 Experiments

This section presents the experiments conducted for this study. First, we describe the duplicate-free splits proposed for the AOLP and CCPD datasets. Then, we list the six OCR models explored in this chapter's assessments. Afterward, we show some examples of the synthetic images created to avoid overfitting during model training. Finally, we report and analyze the results obtained.

[23] CCPD's latest version is available at `https://github.com/detectRecog/CCPD/`

[24] We replicated the split made in (Laroca et al., 2021b) of AOLP's images into training, validation and test sets.

(a) Subset AC (b) Subset LE (c) Subset RP

(d) Subset AC (e) Subset AC (f) Subset RP

Figure 8.2: Examples of images from different subsets in the AOLP dataset that show the same vehicle/LP. In the split protocols often adopted in the literature, some of these images are in the training set and others are in the test set. We show a zoomed-in version of the rectified LP in the lower left region of each image for better viewing.

Subset Base Subset Base Subset Base Subset Base

(a) Training set

Subset Challenge Subset Challenge Subset Weather Subset Weather

(b) Test set

Figure 8.3: The same vehicle/LP may appear in both training and test images in the CCPD dataset (Xu et al., 2018). We show a zoomed-in version of the rectified LP in the lower left region of each image for better viewing.

8.2.1 Duplicate-Free Splits for the AOLP and CCPD Datasets

As the AOLP and CCPD datasets do not have data scraped from the internet (as CIFAR-10 and CIFAR-100 do, for example), we cannot replace the duplicates with new images due to the risk of selection bias or domain shift (Torralba and Efros, 2011; Tommasi et al., 2017; Barz and Denzler, 2020). Therefore, we present ***fair splits*** for each dataset where there are no duplicates of the test images in the training set[25]. As detailed next, we attempted to preserve the key characteristics of the original splits in the new ones as much as possible.

The *AOLP-Fair-A* split was created as follows. Following previous works (Xie et al., 2018; Zhuang et al., 2018; Liang et al., 2022), we randomly divided each of the three subsets of the AOLP dataset into training and test sets with a 2:1 ratio. Nevertheless, we ensured that distinct images showing the same vehicle/LP (as those shown in Figure 8.2) were all in the same set. Afterward, we allocated 20% of the training images for validation. In this way, the AOLP-A (adopted in previous works) and AOLP-Fair-A protocols have the same number of images for training, testing and validation.

The core idea of the AOLP-B protocol is to train the approaches on the AC and LE subsets and test them on the RP subset (Fan and Zhao, 2022; Qin and Liu, 2022; Nguyen, 2022). Thus, we created the *AOLP-Fair-B* protocol in the following way. We kept the original training and validation sets and removed the duplicates from the test set; otherwise, one could ask whether a potential drop in recognition rate is solely due to the reduction in the number of training examples available. In other words, the test sets for the AOLP-B and AOLP-B-Fair splits are different, with the AOLP-B-Fair's test set being a duplicate-free subset of the AOLP-B's test set. However, the training and validation sets are exactly the same in both splits.

As mentioned in Section 8.1.1, CCPD's standard split randomly divides the 200k images of the Base subset into training (100k) and validation (100k) sets. All images from the other subsets are used for testing (except Green, which was introduced later and has its own split). In order to maintain such a distribution, we created the *CCPD-Fair* split as follows. The Base subset was divided into training and validation sets with 100k images each, as in the original split. Nevertheless, instead of making this division completely random, we made the training set free of duplicates by allocating all duplicates to the validation set[26]. Similarly, we followed the original split for the Green subset as closely as possible, just reallocating the duplicates from the training set to the validation set. The test set has not changed. In essence, the original and CCPD-Fair splits use the same $\approx$ 157k images for testing but have different images in the training and validation sets (each with $\approx$ 103k images – about 100k from Base and 3k from Green).

8.2.2 OCR Models

This chapter focuses on six of the OCR models used in previous chapters: CNNG (Fan and Zhao, 2022), Holistic-CNN (Špaňhel et al., 2017), Multi-Task (Gonçalves et al., 2018), STAR-Net (Liu et al., 2016), TRBA (Baek et al., 2019), and ViTSTR-Base (Atienza, 2021b). These models were selected based on their performance in prior evaluations. Note that the CCPD dataset lacks annotations for character positions, rendering CR-NET and Fast-OCR unusable for this analysis.

We trained the models using the same frameworks and hyperparameters as in previous chapters (see Section 5.1 for details).

[25] The list of near-duplicates we have found and proposals for fair splits are publicly available for further research at `https://raysonlaroca.github.io/supp/lpr-train-on-test/`

[26] We trained the OCR models with and without duplicates in CCPD-Fair's validation set, which is used for early stopping and choosing the best weights. As the results achieved in the test set were essentially the same, we kept the same number of validation images (100k-103k) as in the original division.

8.2.3 Synthetic Data

It is well-known that (i) LPR datasets usually have a significant imbalance in terms of character classes as a result of LP assignment policies (Gonçalves et al., 2018; Fan and Zhao, 2022) and (ii) OCR models are prone to memorize patterns seen in the training stage (Zeni and Jung, 2020; Garcia-Bordils et al., 2022); this phenomenon was termed *vocabulary reliance* in (Wan et al., 2020). To mitigate the risk of overfitting, we incorporated many synthetic LP images into the training set. We opted to generate these images using templates, mirroring the methodology outlined in the preceding chapter (Section 7.2.1), as this method does not rely on real images. Examples of the LP images generated for this chapter's experiments are shown in Figure 8.4.

Figure 8.4: Some of the many LP images we created to mitigate overfitting. The images in the top row simulate LPs from vehicles registered in the Taiwan region (as in AOLP), while those in the bottom row simulate LPs from vehicles registered in mainland China (as in CCPD).

8.2.4 Results and Discussion

Here, we report the recognition rates reached by the OCR models in each dataset under the original and fair splits[27]. As usual, recognition rate refers to the number of correctly recognized LPs divided by the number of LPs in the test set. Following (Barz and Denzler, 2020), in addition to the recognition rates obtained in the original and fair protocols, we report their differences in terms of absolute percentage points ("Gap") and in relation to the original error ("Rel. Gap"):

$$Rel.\ Gap = \frac{gap}{100\% - acc} \tag{8.1}$$

The results reached by all OCR models on the AOLP dataset are shown in Tables 8.1 and 8.2. In both protocols (AOLP-A and AOLP-B), the recognition rates obtained in the fair split were considerably lower than those achieved in the original one. Specifically, *the error rates were more than twice as high in the experiments conducted under the fair protocols*.

Table 8.1: Recognition rates achieved by six OCR models under the AOLP-A (adopted in previous works) and AOLP-Fair-A (ours) protocols. The best value in each column is shown in bold.

Model	AOLP-A ↑	AOLP-A-Fair ↑	Gap ↓	Rel. Gap ↓
CNNG (Fan and Zhao, 2022)	98.88%	95.63%	3.25%	290.2%
Holistic-CNN (Špaňhel et al., 2017)	96.75%	93.11%	3.64%	**112.0%**
Multi-Task (Gonçalves et al., 2018)	97.33%	93.79%	3.54%	132.6%
STAR-Net (Liu et al., 2016)	98.69%	95.83%	2.86%	218.3%
TRBA (Baek et al., 2019)	**99.18%**	**96.94%**	2.24%	273.2%
ViTSTR-Base (Atienza, 2021b)	98.74%	**96.94%**	**1.80%**	142.9%

It is crucial to note that the ranking of the recognition models *changed* when they were trained and tested under fair splits. For example, the CNNG model achieved the best result under

[27] We reinforce that all results reported in this chapter are from our experiments (i.e., we trained all recognition models following precisely the same protocol in each set of experiments) and not replicated from the cited papers.

Table 8.2: Recognition rates achieved by six OCR models under the AOLP-B (adopted in previous works) and AOLP-Fair-B (ours) protocols. The best value in each column is shown in bold.

Model	AOLP-B ↑	AOLP-B-Fair ↑	Gap ↓	Rel. Gap ↓
CNNG (Fan and Zhao, 2022)	**98.91%**	96.80%	2.11%	193.6%
Holistic-CNN (Špaňhel et al., 2017)	98.42%	96.30%	2.12%	134.2%
Multi-Task (Gonçalves et al., 2018)	98.42%	95.29%	3.13%	198.1%
STAR-Net (Liu et al., 2016)	98.47%	96.46%	2.01%	131.4%
TRBA (Baek et al., 2019)	98.75%	**97.47%**	**1.28%**	**102.4%**
ViTSTR-Base (Atienza, 2021b)	98.75%	97.31%	1.44%	115.2%

the AOLP-B protocol (as in (Fan and Zhao, 2022), where it was proposed) but only reached the third-best result under AOLP-Fair-B. Similarly, the ViTSTR-Base model ranked third under the AOLP-A protocol but tied for first place with TRBA under AOLP-Fair-A.

These results strongly suggest that, in the past, the high fraction of near-duplicates in the splits traditionally adopted in the literature for the AOLP dataset may have prevented the publication and adoption of more efficient LPR models that can generalize as well as other models but fail to memorize duplicates. A similar concern was raised by Barz and Denzler (2020) with respect to the CIFAR-10 and CIFAR-100 datasets.

The results for the CCPD dataset are presented in Table 8.3, with a further breakdown provided in Table 8.4 following established practices in the field (Xu et al., 2018; Chen et al., 2023; Liu et al., 2024b). While the largest drop in recognition rate was 3.64% in the AOLP dataset, the STAR-Net and TRBA models had drops of 5.20% and 4.35% in recognition rate under the CCPD-Fair protocol, respectively. The average recognition rate decreased from 80.3% to 77.6%, with the relative gaps being much smaller than those observed in the AOLP dataset because the recognition rates reached in CCPD were not as high (we note that lower recognition rates were expected for the CCPD dataset, as its creators modified it twice with the specific purpose of making it much more challenging than it was initially).

Table 8.3: Recognition rates achieved by six well-known recognition models on the CCPD dataset under the standard and CCPD-Fair protocols. The best value in each column is shown in bold.

Model	CCPD ↑	CCPD-Fair ↑	Gap ↓	Rel. Gap ↓
CNNG (Fan and Zhao, 2022)	**88.24%**	**86.93%**	1.31%	11.1%
Holistic-CNN (Špaňhel et al., 2017)	77.01%	75.41%	1.60%	7.0%
Multi-Task (Gonçalves et al., 2018)	83.01%	81.84%	**1.17%**	**6.9%**
STAR-Net (Liu et al., 2016)	78.53%	73.33%	5.20%	24.2%
TRBA (Baek et al., 2019)	75.83%	71.48%	4.35%	18.0%
ViTSTR-Base (Atienza, 2021b)	79.06%	76.37%	2.69%	12.9%

Examining the absolute number of errors may give a clearer understanding of the impact of duplicates on the evaluation of the recognition models. The lowest performance gap of 1.17% translates to 1,800+ additional LPs being misrecognized under the fair split (vs. the standard one), while the highest performance gap of 5.2% represents a staggering number of 8,000+ more LPs being incorrectly recognized under the fair split.

In contrast to the observed in the AOLP dataset, the model rankings remained largely consistent in CCPD, with only the fourth and fifth places switching positions. This is partially due to the significant performance gap between the models and suggests that the community's research efforts have not *yet* overfitted to the presence of duplicates in the standard split of the CCPD dataset. However, we fundamentally believe it is only a matter of time before this starts to happen or be noticed (potentially with the use of deeper models, as the ability to memorize

Table 8.4: Recognition rates (%) for each subset of the CCPD dataset under the standard and CCPD-Fair protocols.

Model \ Subset	Blur 21K	Chal. 50K	DB 10K	FN 21K	Green 5K	Rot. 10K	Tilt 30K	Weath. 10K	All 157K
CCPD									
CNNG (Fan and Zhao, 2022)	77.3	84.1	80.8	91.0	94.2	97.4	95.5	99.3	**88.2**
Holistic-CNN (Špaňhel et al., 2017)	52.0	68.8	67.8	81.9	93.0	95.2	91.4	99.1	77.0
Multi-Task (Gonçalves et al., 2018)	68.4	77.1	73.2	86.1	93.8	96.0	92.6	98.8	83.0
STAR-Net (Liu et al., 2016)	58.7	71.2	64.9	83.3	91.7	94.9	91.2	98.4	78.5
TRBA (Baek et al., 2019)	50.2	67.9	59.6	81.9	92.7	94.7	91.1	98.4	75.8
ViTSTR-Base (Atienza, 2021b)	56.4	72.0	65.9	84.6	94.0	95.5	92.2	98.8	79.1
CCPD-Fair									
CNNG (Fan and Zhao, 2022)	73.4	82.8	78.8	90.2	92.8	97.0	95.1	99.2	**86.9**
Holistic-CNN (Špaňhel et al., 2017)	47.9	66.8	65.6	81.2	91.2	95.1	90.9	98.2	75.4
Multi-Task (Gonçalves et al., 2018)	65.7	75.7	71.5	85.3	92.0	95.6	92.2	98.7	81.8
STAR-Net (Liu et al., 2016)	46.4	64.3	57.2	79.7	91.5	93.9	89.6	98.0	73.3
TRBA (Baek et al., 2019)	38.7	62.7	52.4	80.0	91.2	93.8	89.3	98.1	71.5
ViTSTR-Base (Atienza, 2021b)	50.2	68.4	63.5	82.5	93.5	95.1	91.1	98.7	76.4

training data increases with the model's capacity (Barz and Denzler, 2020; Hooker et al., 2020)) in case such near-duplicates in the training and test sets are not acknowledged and therefore avoided.

8.3 What About Other Datasets?

As mentioned earlier, we focused our analysis on the AOLP and CCPD datasets due to their predominance in the ALPR literature (Xie et al., 2018; Qin and Liu, 2020; Zhang et al., 2020c; Liang et al., 2022; Pham, 2023). Nevertheless, as this issue (i.e., LPR models being evaluated in datasets containing near-duplicates in the training and test sets) has not yet received due attention from the community, it has recurred in assessments carried out on several other public datasets.

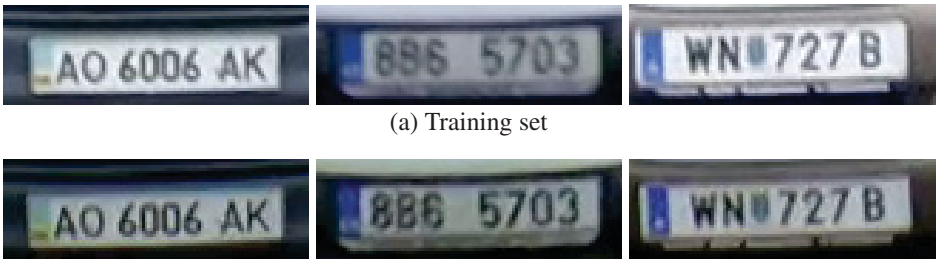
Consider the EnglishLP (Srebrić, 2003), Medialab LPR (Anagnostopoulos et al., 2008) and PKU (Yuan et al., 2017) datasets as examples (they are quite popular, albeit far less than AOLP and CCPD). They all have near-duplicates, as shown in Figure 8.5. As these datasets lack an official evaluation protocol, it is common for authors to divide their images into training, validation and test sets randomly (Zhuang et al., 2018; Gao et al., 2020a; Khan et al., 2021; Zhang et al., 2021d; Qin and Liu, 2022). As can be inferred, the presence of near-duplicates in these datasets has also been overlooked in such setups.

The ReId dataset (Špaňhel et al., 2017) differs from the datasets mentioned above by having a standard protocol. It has 182,335 images of cropped low-resolution LPs, of which 105,923 are in the training set and 76,412 are in the test set. We found that 52,394 (**68.6%**) of the test images have near-duplicates in the training set (see some examples in Figure 8.6). Although alarming, the high fraction of duplicates has gone unacknowledged in works using the ReId dataset for experimentation (Špaňhel et al., 2018; Wu et al., 2019; Moussa et al., 2022).

We also want to draw attention to the fact that there are duplicates even across different datasets. Recently, Zhang et al. (2021c) released the CLPD dataset, which comprises 1,200 images gathered from multiple sources such as the internet, mobile phones, and car driving recorders. The authors employed all images for testing to verify the practicality of their LP detection and recognition models, trained on other datasets. Subsequent studies have followed this protocol (Zou et al., 2020, 2022; Liu et al., 2021; Zhang et al., 2021d; Chen et al., 2023; Ke et al., 2023; Rao et al., 2024). The problem is that several vehicles/LPs shown in CLPD are also shown in the ChineseLP dataset (Zhou et al., 2012) (see Figure 8.7). That is, if not yet, images from the ChineseLP dataset will eventually be used to train ALPR systems that will then be tested

(a) EnglishLP (b) Medialab LPR (c) PKU

Figure 8.5: ALPR datasets that do not have a well-defined evaluation protocol are customarily divided into training and test sets randomly without the authors noticing that the same vehicle/LP may appear in multiple images. Above, we show a pair of near-duplicates from each of the EnglishLP, Medialab LPR, and PKU datasets. Observe that it is common for an LP to look very similar in different images even without rectification. We show a zoomed-in version of the rectified LP in the lower left region of each image for better viewing.



(a) Training set

(b) Test set

Figure 8.6: Examples of near-duplicates in the ReId dataset (Špaňhel et al., 2017). It is clear that such duplicates may also considerably bias the evaluation of ALPR systems that do not perform rectification before the LPR stage.

on the CLPD dataset. These experiments will likely be regarded as "cross-dataset," although perhaps they should not. In our experiments, presented in previous chapters, we addressed this concern by excluding from the training set any images from the ChineseLP dataset that are also present in the CLPD dataset (see footnote [18] on page 93).

One last example that highlights the overlooked nature of this issue can be found in the work of Gong et al. (2022). They presented a detailed comparison between multiple datasets gathered in mainland China, including ChineseLP and CLPD, without noticing the existence of duplicates across them.

It is noteworthy that we incorporated measures while defining the standard split for the RodoSol-ALPR dataset to ensure the absence of duplicates within the training and test sets.

8.4 Final Remarks

We drew attention to the large fraction of near-duplicates within the training and test sets of datasets widely adopted in ALPR research. Both the existence of such duplicates and their influence on the performance evaluation of LPR models have largely gone unnoticed in the literature.

Our experiments on the AOLP and CCPD datasets, the most commonly used in the field, showed that the presence of near-duplicates significantly impacts the performance evaluation

(a) Images from ChineseLP (Zhou et al., 2012)

(b) Images from CLPD (Zhang et al., 2021c)

Figure 8.7: There are duplicates even across different datasets. The above images were taken from the ChineseLP and CLPD datasets, both of which contain images scraped from the internet. The presence of near-duplicates across datasets can significantly bias the results of cross-dataset experiments.

of OCR models applied to LPR. In the AOLP dataset, the error rates reported by the models were more than twice as high in the experiments conducted under the fair splits. The ranking of the models also changed when they were trained and tested under duplicate-free splits. In the more challenging CCPD dataset, the models showed recognition rate drops of up to 5.2%. Specifically, the average recognition rate decreased from 80.3% to 77.6% when the experiments were conducted under the fair split compared to the standard one. These results indicate that duplicates have biased the evaluation and development of deep learning-based models for LPR.

We created the *fair splits* for the abovementioned datasets by dividing their images into new training, validation and test sets while ensuring that no duplicates from the test set are present in the training set and preserving the original splits' key characteristics as much as possible. These new splits and the list of duplicates found are publicly available.

We hope the work conducted in this chapter will encourage LPR researchers to train and evaluate their models using the fair splits we created for the AOLP and CCPD datasets and to beware of duplicates when performing experiments on other datasets. This chapter also provides researchers with a clearer understanding of the true capabilities of LPR models that have only been evaluated on test sets that include duplicates from the training set.

9. A FIRST LOOK AT DATASET BIAS IN LICENSE PLATE RECOGNITION

Is it possible to accurately determine the dataset from which an LP image originates? Initially, one may think that this task is fairly trivial since – in principle – images from distinct datasets are collected in different regions, with different hardware, for different purposes, etc. On second thought, one may realize that it depends on the datasets we are comparing.

Suppose there are two datasets, one composed exclusively of images of American LPs and the other of images of European LPs. In that case, it should indeed be relatively straightforward to distinguish which dataset each LP image belongs to due to the many characteristics LPs from the same region share in common, e.g., the aspect ratio, colors, symbols, the position of the characters, the number of characters, among others. Nevertheless, beyond the LP layout, are there unique signatures (*bias*) in each dataset that would enable identifying the source of an LP image?

The presence of unique signatures in public datasets was first revealed by Torralba and Efros (2011). They investigated the then-popular object recognition datasets (PASCAL'07, ImageNet, among others) using the *Name That Dataset!* experiment in which a Support Vector Machine (SVM) classifier was trained to distinguish images from 12 datasets. If *dataset bias* did not exist, no classifier would be able to perform this task at levels considerably different from chance. However, their classifier reached an accuracy of 39%, which is significantly better than chance ($1/12 = 8\%$). This result becomes even more surprising when taking into account that those datasets were created with the expressed goal of being as varied and rich as possible, aiming to sample the visual world "in the wild" (Torralba and Efros, 2011).

Dataset bias has been consistently recognized as a severe problem in the computer vision community (Tommasi et al., 2017; Ashraf et al., 2018; Wachinger et al., 2021; Jaipuria et al., 2022; Hort et al., 2023), given that models are inadvertently learning idiosyncrasies of each dataset along with knowledge fundamental to the task under study. Nevertheless, to the best of our knowledge, this bias has remained largely unnoticed in the LPR literature.

Considering the above discussion, in this chapter we revisit the experiments conducted by Torralba and Efros (2011), adapting them to the LPR context[28] (see Figure 9.1, where we recreate the *Name That Dataset!* game with Brazilian LPs). Our experiments, performed on public datasets acquired in Brazil and mainland China, demonstrate that a lightweight CNN can identify the source dataset of an LP image with more than 95% accuracy, which is much higher than expected from chance or human perceptual similarity judgments. Intriguingly, our experiments also show no signs of saturation as more training data is added, i.e., the classification accuracy could be even higher if there were more training data.

The severity of the dataset bias problem in LPR boils down to the following. LPR datasets are usually very unbalanced in terms of character classes due to LP assignment policies, as previously discussed. In a dataset collected in Brazil, for instance, one letter may appear much more frequently than others according to the state in which most vehicles were registered; for example, the SSIG-SegPlate dataset (Gonçalves et al., 2016a) has 746 instances of the letter 'O' but only 135 instances of the letter 'Q'. The same is true for vehicles registered in different cities within a province in mainland China (Zhang et al., 2021c; Wang et al., 2022c). Taking into account that LPR models are generally trained and evaluated on images from the same dataset (as detailed in Chapter 5), such bias can skew the predictions toward the prominent character classes

[28] This chapter – in article form – was accepted for presentation at the *2022 Conference on Graphics, Patterns and Images (SIBGRAPI)* (Laroca et al., 2022b).

Figure 9.1: Can you name the dataset to which each of the above images belongs? (you can try grouping the images into four distinct groups if you are unfamiliar with the corresponding datasets). See footnote[29] for the answer key. This task is somewhat challenging for humans, as LP images from distinct datasets have similar characteristics. However, a shallow CNN (3 conv. layers) predicts the correct dataset in more than 95% of cases (chance is $1/4 = 25\%$). All images above were classified correctly, with a mean confidence value of 95.9%.

within that particular dataset, resulting in poor performance on other datasets and, naturally, in real-world scenarios (Yang et al., 2018; Zhang et al., 2020c).

The aim of this chapter is two-fold. First, to situate the dataset bias problem in the LPR context and thus raise awareness in the community regarding the possible impacts of such bias as this issue is not getting the attention it deserves. Second, to discuss some subtle ways bias may have crept into the chosen datasets to outline directions for future research.

The subsequent sections of this chapter are structured as follows. Section 9.1 provides a concise overview of the motivation behind this chapter. Section 9.2 outlines the experiments carried out and presents the corresponding results. In Section 9.3, we shed light on the impacts of dataset bias on the cross-dataset generalization of LPR models, offering insights into potential causes. Lastly, Section 9.4 summarizes the key findings of the chapter.

9.1 Motivation

The standard method of evaluating an LPR method's performance is to use multiple publicly available datasets, such as SSIG-SegPlate (Gonçalves et al., 2016a) and CCPD (Xu et al., 2018), which are split into disjoint training and test sets. Such an assessment is typically done independently for each dataset (Zhuang et al., 2018; Weihong and Jiaoyang, 2020; Zhang et al., 2021d; Ke et al., 2023). As models based on deep learning can take significant time to be trained, some authors have adopted a slightly different protocol where the proposed networks are trained once on the union of the training images from the chosen datasets and evaluated individually on the respective test sets (Selmi et al., 2020; Laroca et al., 2021b; Qin and Liu, 2022; Silva and Jung, 2022). Although the images for training and testing belong to disjoint subsets, these protocols do not make it clear whether the evaluated models have good generalization ability, i.e., whether they perform well on images from other scenarios/datasets, mainly due to domain divergence and data selection bias (Torralba and Efros, 2011; Tommasi et al., 2017).

In Chapter 5, we showed that there are significant drops in LPR performance across various datasets when employing well-known OCR models such as Facebook's Rosetta (Borisyuk

[29] Answer key: RodoSol-ALPR → (a),(d),(h),(l); SSIG-SegPlate → (e),(i),(j),(o); UFOP → (b),(f),(m),(n); and UFPR-ALPR → (c),(g),(k).

et al., 2018) and TRBA (Baek et al., 2019) in a leave-one-dataset-out experimental setup. Initially, we attributed such underwhelming results to the heavy bias toward specific regional identifiers within existing datasets for LPR. Nevertheless, we employed a large volume of synthetic data (generated through character permutation) to mitigate such bias during those experiments. This led us to hypothesize that there are *other* strong biases crept into LPR datasets. This realization serves as the primary motivation for the research presented in this chapter.

9.2 Experiments

This section describes the experiments performed in this work. We first list the datasets explored in our assessments, explaining why they were chosen and not others. We also detail how the LP images from each dataset were selected and divided into training, validation and test subsets. Then, we describe the CNN model employed for the dataset classification task (*Name That Dataset!* game) and provide implementation details. Finally, we report the results achieved.

9.2.1 Datasets

Our experiments were carried out on images from eight public datasets introduced over the last decade: RodoSol-ALPR, SSIG-SegPlate (Gonçalves et al., 2016a), UFOP (Mendes Júnior et al., 2011), UFPR-ALPR (Laroca et al., 2018), a reduced version of CCPD (Xu et al., 2018), ChineseLP (Zhou et al., 2012), PKU (Yuan et al., 2017), and PlatesMania-CN (Laroca et al., 2021b). The images of the first four datasets were acquired in three states of Brazil, while the images of the last four datasets were collected in various provinces of mainland China. We cropped the LP regions from the original images (taken in urban environments) for our experiments.

In this chapter, we chose to experiment with LPs from Brazil and mainland China because there are many ALPR systems designed primarily for LPs from one of those regions (Silva and Jung, 2017; Silvano et al., 2021; Gong et al., 2022; Jiang et al., 2023b). Considering the objectives of our study, we also filter which LP images from each dataset to use in our experiments: (i) regarding the datasets collected in Brazil, we explore only LPs that have a single row of characters and gray as the background color (LPs for private vehicles before the implementation of the Mercosur standard); and (ii) for the datasets acquired in mainland China, we explore only LPs that have a single row of characters and blue as the background color. This protocol was adopted because the four datasets collected in each region have LPs with these characteristics. In contrast, only some datasets have LPs with other characteristics (e.g., UFOP and SSIG-SegPlate do not have any two-row LPs, and the ChineseLP and PlatesMania-CN datasets do not include LPs with yellow background). An overview of the datasets used in our experiments, after the aforementioned selection process, is presented in Table 9.1. We labeled the color of each LP in every dataset to make this selection, and these annotations are publicly available[30].

For reproducibility, it is essential to make clear how we divided the selected images from each of the datasets to train, validate and test the classification model (detailed in Section 9.2.2). The CCPD, RodoSol-ALPR, SSIG-SegPlate and UFPR-ALPR datasets were split according to the protocols defined by the respective authors (i.e., the authors specified which images belong to which subsets), while the other datasets, which do not have well-defined evaluation protocols, were randomly split into 40% images for training; 20% images for validation; and 40% images for testing, following the split protocol adopted in the SSIG-SegPlate and UFPR-ALPR datasets[31].

[30] https://raysonlaroca.github.io/supp/sibgrapi2022/annotations.zip

[31] The training, validation, and test splits are available at https://raysonlaroca.github.io/supp/sibgrapi2022/splits.zip

Table 9.1: Datasets used for the experiments conducted in this chapter.

Dataset	Year	LP Images	State / Province-City
UFOP	2011	244	Minas Gerais (BR)
ChineseLP	2012	400	Various (CN)
SSIG-SegPlate	2016	1,832	Minas Gerais (BR)
PKU	2017	2,024	Anhui-Tongling (CN)
UFPR-ALPR	2018	2,700	Paraná (BR)
CCPD	2020*	25,000†	Anhui-Hefei (CN)
PlatesMania-CN	2021	347	Various (CN)
RodoSol-ALPR	2022	4,765	Espírito Santo (BR)

* The CCPD dataset was introduced in 2018 and last updated in 2020.
† Following (Liu et al., 2021), we used a reduced version of CCPD in our experiments.

As the CCPD dataset has many more images than the others (more than 350k), we followed (Liu et al., 2021) and performed our experiments using a reduced version with 25k images.

Three points should be noted. First, for all datasets, we were careful not to have images of the same LP in different subsets (otherwise, different images of an LP could appear in both the training and test sets, for example). Second, as the chosen datasets have different numbers of test images, we randomly sample a set of N test set images from different datasets to predict which dataset each image belongs to (for each region, N is constrained by the smallest number of images in the test sets). This experiment is repeated 100 times with different splits and we report the average results. Similar protocols were adopted in (Torralba and Efros, 2011; Khosla et al., 2012; Tommasi et al., 2017). Third, as in other chapters of this thesis, we used Albumentations (Buslaev et al., 2020) to balance the number of training images from different datasets, thus mitigating overfitting. Transformations applied to generate new images include random noise, random JPEG compression, random shadows, and random perturbations of hue, saturation and brightness.

For clarity, throughout the remainder of this chapter, "Brazilian LPs" refer to gray single-row LPs from vehicles registered in Brazil (prior to the adoption of the Mercosur layout), and "Chinese LPs" refer to blue single-row LPs from vehicles registered in mainland China. While some examples of Brazilian LPs can be seen in Figure 9.1 (the teaser image of this chapter), some Chinese LPs from the chosen datasets are shown in Figure 9.2.

Figure 9.2: Some Chinese LPs from the datasets used in this chapter. From top to bottom: CCPD (Xu et al., 2018), ChineseLP (Zhou et al., 2012), PKU (Yuan et al., 2017) and PlatesMania-CN (Laroca et al., 2021b).

One may have noticed that all LP images we showed (both in Figure 9.1 and Figure 9.2) are quite horizontal, tightly bounded, and "easy" to read. This is because we rectified all LPs to eliminate biases such as repetitive tilt angles caused by specific camera positions in images from distinct datasets. To perform the rectification, we labeled the position (x, y) of the four corners of each LP in the eight datasets that do not contain such labels (only the CCPD and RodoSol-ALPR datasets have corner annotations for all LPs). These newly created annotations are also accessible at the URL referenced in footnote [30] on the previous page.

9.2.2 Classification Model

For the dataset classification task (*Name That Dataset!*), we designed a lightweight CNN architecture called DC-NET. It is inspired by the CDCC-NET model (Laroca et al., 2021a) and is relatively similar to the model used for this same task in (McLaughlin et al., 2015).

DC-NET's architecture is shown in Table 9.2. As can be seen, the model is relatively shallow, with three convolutional layers containing 16/32/64 filters, each followed by a max-pooling layer with a 2×2 kernel and stride = 2. Batch normalization, followed by a Rectified Linear Unit (ReLU), is added after each convolutional layer. We evaluated several changes to this architecture, such as using depthwise separable convolutional layers, convolutional layers with stride = 2 (removing the max-pooling layers), and different input sizes and numbers of filters. However, better results were not obtained (we conducted these experiments in the validation set).

Table 9.2: DC-NET's layers and hyperparameters.

#	Layer	Filters	Size / Stride	Input	Output
0	conv	16	$3 \times 3/1$	$192 \times 64 \times 3$	$192 \times 64 \times 16$
1	max		$2 \times 2/2$	$192 \times 64 \times 16$	$96 \times 32 \times 16$
2	conv	32	$3 \times 3/1$	$96 \times 32 \times 16$	$96 \times 32 \times 32$
3	max		$2 \times 2/2$	$96 \times 32 \times 32$	$48 \times 16 \times 32$
4	conv	64	$3 \times 3/1$	$48 \times 16 \times 32$	$48 \times 16 \times 64$
5	max		$2 \times 2/2$	$48 \times 16 \times 64$	$24 \times 8 \times 64$
6	flatten			$24 \times 8 \times 64$	12288
#	**Layer**		**Units**	**Input**	**Output**
7	dense		128	12288	128
8	dense		4	128	4

The DC-NET model was implemented using Keras. We used the Adam optimizer, initial learning rate = 10^{-3} (with *ReduceLROnPlateau*'s patience = 3 and factor = 10^{-1}), batch size = 64, max epochs = 50, and patience = 7. In our test environment, equipped with an NVIDIA Quadro RTX 8000 GPU as described in preceding chapters, DC-NET runs at approximately 720 FPS.

9.2.3 Results

In this subsection, we report the results obtained by DC-NET in the dataset classification task (*Name That Dataset!*). Figure 9.3 shows the confusion matrices for Brazilian (left) and Chinese (right) LPs. There is a clearly pronounced diagonal in both matrices, indicating that *each dataset does have a unique, identifiable "signature"*; it is worth noting that only about 25% accuracy would be expected if the classifier was operating at chance levels, as would happen if the LP images from each dataset were fully unbiased samples. The overall accuracy was 95.2% for Brazilian LPs and 95.9% for Chinese LPs.

The results show that the DC-NET model is more successful in classifying LP images from the datasets acquired with static cameras (RodoSol-ALPR, SSIG-SegPlate, UFOP, and PKU) than LP images from the datasets captured by handheld (CCPD, ChineseLP, and PlatesMania-CN) or moving cameras (UFPR-ALPR). We believe this is because images collected by static cameras have many characteristics in common, not just the background. These similarities likely extend to the LP regions, explaining the model's greater accuracy with such images. To illustrate, in Figure 9.4, we show two pairs of the most similar images – in terms of Mean Squared Error (MSE) – from distinct subsets from each of the RodoSol-ALPR and UFPR-ALPR datasets (the datasets where the highest and worst accuracy were achieved, respectively). Observe that factors common in images taken by static cameras, such as similar vehicle positioning and

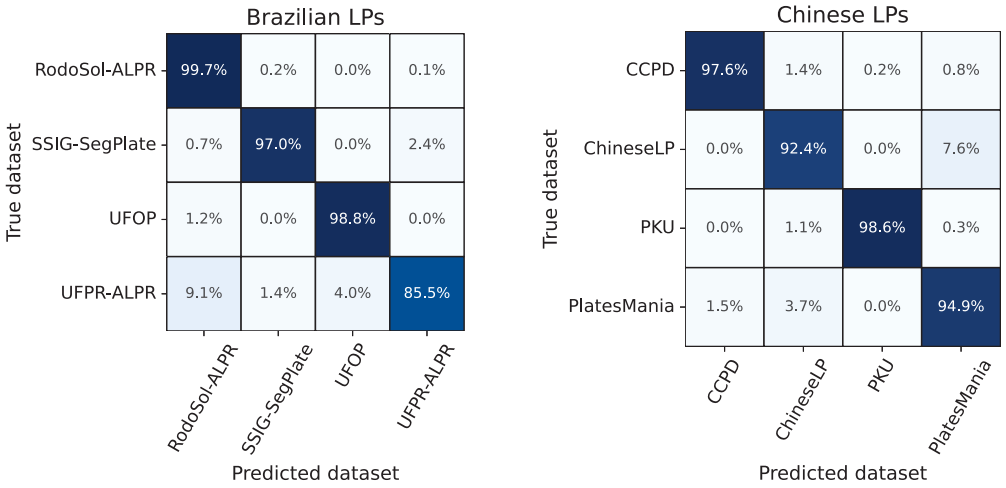


Figure 9.3: Confusion matrices for a classifier (DC-NET) trained to predict the source dataset of a given LP image. Left: Brazilian LPs; right: Chinese LPs.

distance from the camera, may cause the LPs from different images to be quite resembling (note that this is not always the case; it may seem so because we focused on the *most* similar pairs of images from these datasets for this analysis).

(a) MSE = 174 (b) MSE = 407

(c) MSE = 1,686 (d) MSE = 1,700

Figure 9.4: Two pairs of the most similar images (in terms of MSE) from distinct subsets from each of the RodoSol-ALPR (a, b) and UFPR-ALPR (c, d) datasets. In each pair, the left image belongs to the training set, while the right one belongs to the test set. Observe that LPs from different images captured by static cameras may be quite resembling. We show a zoomed-in version of the LP in the lower left region of each image for better viewing.

One might initially suspect the model simply memorized the most frequent regional characters in each dataset (e.g., most LPs in the CCPD dataset have '皖' as the first character). However, this does not hold since DC-NET correctly classified more than 97% of the LP images from both datasets collected in the Brazilian state of Minas Gerais (SSIG-SegPlate and UFOP) and from both datasets acquired in the Anhui province in mainland China (CCPD and PKU).

By carefully analyzing the confusion matrices in Figure 9.3, we noticed that almost all incorrect predictions on Chinese LPs were between the ChineseLP and PlatesMania-CN datasets. We consider this occurred because both datasets have images collected from the internet (the other six datasets do not contain any images from the internet). Specifically, all images from the PlatesMania-CN datasets were downloaded from the internet (Laroca et al., 2021b), while around 39% of the ChineseLP's images were taken from the internet (Zhou et al., 2012). It makes perfect sense that the bias is less pronounced when the images come from multiple sources. The classifier still managing to achieve high accuracy rates in both datasets is due to *selection*

bias (Torralba and Efros, 2011; McLaughlin et al., 2015; Wachinger et al., 2021), which arises because authors building a dataset select images with specific purposes in mind, thus reducing the variability of the data (in many cases without even realizing it). Furthermore, these datasets have images with different quality levels, as they were introduced years apart and the capture devices evolved considerably in the time between them being collected.

Torralba and Efros (2011) observed that using more training data led to higher accuracy, without any immediate signs of saturation. Intrigued by these findings, we trained the DC-NET model three more times for each LP layout: using 50%, 25% and 12.5% of the training data (randomly selected). As depicted in Figure 9.5, our experiments corroborate this trend: the accuracy improves as the size of the training set increases, with no signs of saturation yet observed.

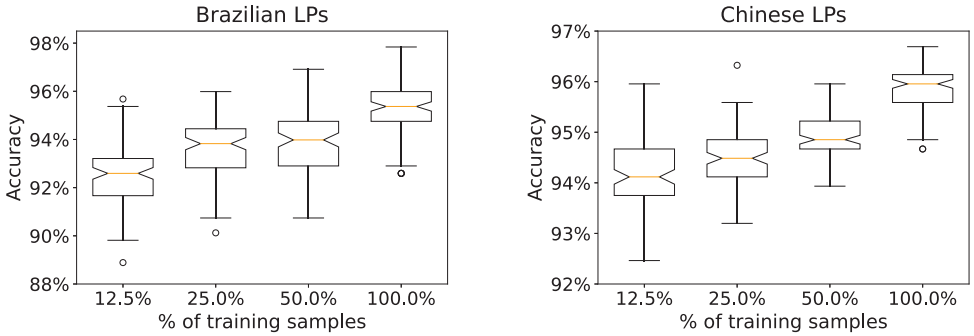


Figure 9.5: Classification performance as a function of training data size. The performance does not seem to be saturated for either Brazilian (left) or Chinese (right) LPs.

Another noteworthy finding is that the classifier predicts the source dataset correctly with a significantly higher confidence value than when it predicts incorrectly. The mean confidence values for correctly classified Brazilian and Chinese LPs were 98.5% and 98.1%, respectively, while the mean confidence values for incorrectly classified Brazilian and Chinese LPs were 79.7% and 74.3%, respectively. Figure 9.6 shows the Receiver Operating Characteristic (ROC) curves for Brazilian (left) and Chinese (right) LPs. Since ROC curves are typically used in binary classification, we binarized the classifier's output (per class) to draw one ROC curve per dataset.

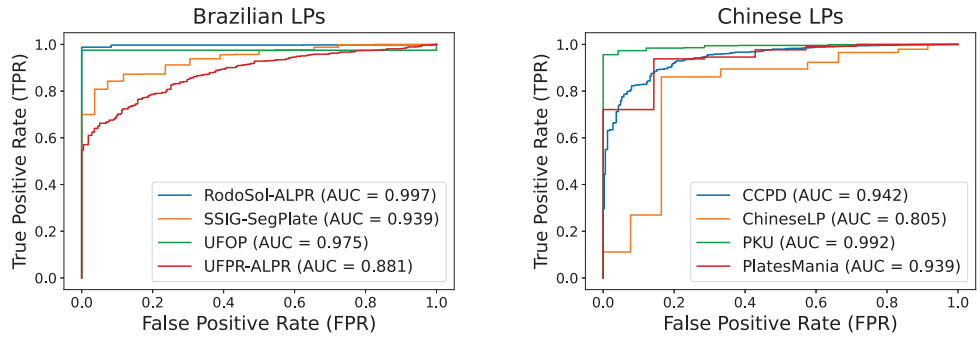


Figure 9.6: ROC curves for Brazilian and Chinese LPs. Note the high Area Under the Curve (AUC) values, which indicate that DC-NET performs considerably well at distinguishing between LP images from different datasets.

9.3 Discussion

Considering that the DC-NET model – which is relatively shallow – can predict the source dataset of an LP image with accuracy above 95%, we conjecture that most LPR models – which are considerably deeper – are actually learning and exploiting such signatures to improve the results

achieved in seen datasets at the cost of losing generalization capability. The intuition behind this conjecture is as follows: consider the SSIG-SegPlate dataset (Gonçalves et al., 2016a) as an example, it has many LPs with the letter 'O' as the first character but no LP with the letter 'Q' in that position. Hence, an LPR model capable of identifying that a given LP image belongs to this dataset may predict the letter 'O' as the first character even if the character looks more like 'Q' than 'O' due to noise or other factors. However, the relatively high recognition rates achieved in the SSIG-SegPlate dataset would likely not be reached in unseen datasets.

This chapter's findings echo the concerns raised in Chapter 5, where we observed significant drops in recognition performance across several datasets when training and testing the models in a leave-one-dataset-out (LODO) fashion. It is important to recall our earlier observation in that chapter (more specifically, in Figure 5.5), where we emphasized that errors under the LODO protocol were not not primarily associated with challenging scenarios, suggesting that they likely stemmed from *differences between the training and testing data distributions*.

We believe that the main cause of dataset bias is related to the cameras used to collect the images in each dataset. Taking the results achieved in Brazilian LPs as an example, the lowest accuracy (i.e., less pronounced bias) was reported for the UFPR-ALPR dataset, which was captured by three non-static cameras of different price ranges. In contrast, the other datasets have images acquired by a single static camera (SSIG-SegPlate and UFOP) or by multiple static cameras of the same model (RodoSol-ALPR). In the same direction, another probable cause of bias relates to how the images were stored in different datasets. For example, the CCPD dataset contains highly compressed images while most other datasets do not. DC-NET probably exploited the detection of artifacts in the highly compressed LP images for better classification.

Some works have linked dataset bias to image backgrounds (McLaughlin et al., 2015; Tian et al., 2018). For example, a classifier may accurately classify images labeled as "boat" without actually focusing on the boat itself, but rather on the water below or the shore in the distance (Torralba and Efros, 2011). Although we are convinced that we have eliminated such bias by performing our experiments on rectified LP images, it is worth noting that the corner annotations in the CCPD dataset are not as accurate as those we made or those found in other datasets. The DC-NET model may have exploited these subtle distinctions as well.

While these conclusions have been reached for the particular classifier used in our experiments, similar trends are expected to hold for similar models (McLaughlin et al., 2015).

We consider two initial ways to mitigate the dataset bias problem in LPR. The first is leveraging deep learning-based methods' high capability to visualize and understand how bias has crept into the chosen datasets. One technique that immediately comes to mind is Grad-CAM (Selvaraju et al., 2017), which uses the gradients of any target class flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the class.

The other way is to embrace the "wildness" of the internet to collect a large-scale dataset for LPR. However, as shown in Section 9.2 and in (Torralba and Efros, 2011), downloading images from the internet alone does *not* guarantee a bias-free sampling, as keyword-based searches return only particular types of images; users of a specific website prefer images with certain characteristics, among other factors. Thus, such a dataset should be obtained from multiple sources on the internet (e.g., multiple search engines and websites from various countries).

9.4 Final Remarks

In this chapter, we situated the dataset bias problem (Torralba and Efros, 2011; Tommasi et al., 2017) in the LPR context. We performed experiments on LP images from eight publicly available

datasets; four were collected in Brazil and four in mainland China. The results showed that each dataset does have a unique, identifiable signature.

Specifically, our *Name That Dataset!* experiments showed that the source dataset of an LP image could be predicted with more than 95% accuracy (chance is $1/4 = 25\%$). Intriguingly, we observed no evidence of saturation as more training data was added. We believe there is no theoretical reason for such results other than the strong biases in the actual datasets.

We hope these findings will further encourage the evaluation of LPR models in cross-dataset setups, as they provide a better indication of generalization (hence real-world performance) than intra-dataset ones.

10. CONCLUSIONS AND FUTURE DIRECTIONS

This thesis contributes significantly to the advancement of Automatic License Plate Recognition (ALPR) by identifying and addressing key limitations in the existing literature.

We tackled the lack of attention given to images featuring Mercosur LPs, motorcycles, and two-row LPs by creating a dedicated dataset (Chapter 4) and conducting many experiments on it (Chapters 5 to 7 and 9). In Chapter 5 specifically, we showed the importance of the RodoSol-ALPR dataset for robust recognition of Mercosur and two-row LPs, as none of the OCR models we trained surpassed a 70% recognition rate on its test set under the leave-one-dataset-out protocol.

In Chapters 6 and 7, we demonstrated that significant improvements in ALPR results could indeed be attained without relying on additional real training data, groundbreaking descriptor designs, or extensive searches for better model architectures. Chapter 6 examined the potential enhancements in LPR results by fusing the outputs from multiple OCR models using straightforward approaches such as selecting the most confident prediction or through majority voting. Chapter 7 explored the synergistic benefits of combining various synthetic data generation methodologies not only to improve LPR performance but also to overcome challenges posed by limited training data availability. Notably, both chapters detailed the enhancements achieved in scenarios observed during training (intra-dataset) as well as on entirely new, unseen data (cross-dataset). Moreover, they compared the balance between speed and accuracy across different approaches, recognizing the importance of efficient systems in real-world applications.

By utilizing a traditional-split vs. leave-one-dataset-out experimental setup, we identified a critical issue in the way ALPR systems have been evaluated. Specifically, the established protocols for assessing these systems have historically failed to accurately indicate their out-of-domain robustness. Our investigation in Chapter 8 revealed that these protocols were formulated without accounting for instances where the same vehicle or LP appears in multiple images. This resulted in many near-duplicates within the training and test sets of the two most referenced datasets in the field, potentially hindering the development and acceptance of more efficient LPR models that have strong generalization abilities but do not memorize duplicates as well as other models. Furthermore, Chapter 9 contextualized the dataset bias problem within the LPR domain. We discovered that OCR models are inadvertently learning idiosyncrasies of each dataset alongside fundamental LPR-related knowledge. All these findings underscore the importance of conducting cross-dataset experiments, as they provide a better indication of generalization (hence real-world performance) than intra-dataset ones. In other words, the outcomes from cross-dataset experiments are more likely to reflect what would be observed in real-world deployments.

Future Directions

Regarding improving the out-of-domain robustness of OCR models applied to LPR, a promising avenue for future research lies in leveraging *adversarial training*. This approach entails incorporating carefully crafted adversarial examples – inputs specifically designed to mislead the models – into the training data. Several studies have shown that adversarial training not only enhances the performance of deep learning models against unforeseen attacks but also boosts their accuracy on both clean images and out-of-domain samples (Zhao et al., 2020a; Poursaeed et al., 2021; Lehner et al., 2024). Despite these potential benefits, the exploration of adversarial training within the ALPR domain remains largely unexplored.

Beyond exploring adversarial training, we firmly believe that significantly improved results can be attained with minimal manual effort by utilizing *coarse annotations*. These annotations can be automatically generated for unlabeled images from the internet or public

datasets that either lack annotations entirely or only have labels for specific parts of the ALPR pipeline. While research across various domains demonstrates the effectiveness of coarse annotations in boosting deep learning model performance (Lucio et al., 2019; Liu et al., 2020; Das et al., 2023), there is still a gap in understanding how to best couple coarse-annotated data with fine-annotated data in the ALPR context. Such exploration should focus on mitigating annotation errors and their adverse effects on network learning.

After successfully applying synthetic data to accurately recognize LPs on high-quality images captured across various scenarios and regions (Chapter 7), we suggest a progressive shift in focus for ALPR researchers toward tackling the challenges associated with detecting and recognizing *low-quality* and *low-resolution* LPs. These challenges are often encountered in criminal investigations, where video evidence typically comes from security cameras not optimized for ALPR. The LPs in these videos are commonly illegible throughout the entire recording. Possible solutions to this problem include exploring *image enhancement* techniques, such as *super-resolution*, and leveraging temporal information by analyzing *multiple frames*. Current research on LP image enhancement has predominantly focused on unrealistic scenarios, such as synthetic low-resolution images created by artificially downsampling high-resolution ones (Schirrmacher et al., 2023; Kim et al., 2024). Similarly, studies using multiple frames have often relied on basic majority voting from individual frames (Al-batat et al., 2022; Silva and Jung, 2022), failing to utilize the full potential of feeding sequential frames into the models.

Last but not least, exploring how to effectively utilize the high capabilities of deep learning methods to address the *dataset bias* issue in LPR (Chapter 9) remains an open area for future research. As an initial step toward this goal, we advocate exploring visualization techniques such as Grad-CAM (Selvaraju et al., 2017) and Iterated Integrated Attributions (IIA) (Barkan et al., 2023). These techniques can generate visually interpretable heatmaps, offering insights into how a lightweight classifier can excel at distinguishing between LP images from different datasets.

REFERENCES

Aberdam, A., Litman, R., Tsiper, S., Anschel, O., Slossberg, R., Mazor, S., Manmatha, R., and Perona, P. (2021). Sequence-to-sequence contrastive learning for text recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15297–15307.

Akoushideh, A., Shahbahrami, A., and Joe Afshany, A. (2024). Parallelization of license plate localization on GPU platform. *Multimedia Tools and Applications*, 83(1):2551–2564.

Al-batat, R., Angelopoulou, A., Premkumar, S., Hemanth, J., and Kapetanios, E. (2022). An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification. *Sensors*, 22(23):9477.

Al-Shemarry, M. S. and Li, Y. (2020). Developing learning-based preprocessing methods for detecting complicated vehicle licence plates. *IEEE Access*, 8:170951–170966.

Anagnostopoulos, C. N. E. et al. (2006). A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Transactions on Intelligent Transportation Systems*, 7(3):377–392.

Anagnostopoulos, C. N. E. et al. (2008). License plate recognition from still images and video sequences: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):377–391.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 214–223.

arXiv (2024). Monthly submissions. `https://arxiv.org/stats/monthly_submissions`.

Ashraf, A., Khan, S. S., Bhagwat, N., and Taati, B. (2018). Learning to unlearn: Building immunity to dataset bias in medical imaging studies. In *Machine Learning for Health Workshop at NeurIPS 2018*.

Ashrafee, A., Khan, A. M., Irbaz, M. S., and Nasim, M. A. A. (2022). Real-time Bangla license plate recognition system for low resource video-based applications. In *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 479–488.

Atienza, R. (2021a). Data augmentation for scene text recognition. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1561–1570.

Atienza, R. (2021b). Vision transformer for fast and efficient scene text recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 319–334.

Atienza, R. (2022). Vision Transformer for Fast and Efficient Scene Text Recognition. `https://github.com/roatienza/deep-text-recognition-benchmark`.

Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., and Lee, H. (2019). What is wrong with scene text recognition model comparisons? dataset and model analysis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4714–4722.

Baek, J., Matsui, Y., and Aizawa, K. (2021a). 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)*, pages 3112–3121.

Baek, K. et al. (2021b). Rethinking the truly unsupervised image-to-image translation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14154–14163.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, pages 1–15.

Bang, D. and Shim, H. (2021). MGGAN: Solving mode collapse using manifold-guided training. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2347–2356.

Barkan, O., Elisha, Y., Eshel, A., and Koenigstein, N. (2023). Visual explanations via iterated integrated attributions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2073–2084.

Barz, B. and Denzler, J. (2020). Do we train on test data? Purging CIFAR of near-duplicates. *Journal of Imaging*, 6(6):41.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Bengio, Y., Lecun, Y., and Hinton, G. (2021). Deep learning for AI. *Communications of the ACM*, 64(7):58–65.

Beratoğlu, M. S. and Töreyin, B. U. (2021). Vehicle license plate detector in compressed domain. *IEEE Access*, 9:95087–95096.

Bezerra, C. S. et al. (2018). Robust iris segmentation based on fully convolutional networks and generative adversarial networks. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 281–288.

Bhargav, R. and Deshpande, P. (2019). Locating multiple license plates using scale, rotation, and colour-independent clustering and filtering techniques. *IET Image Processing*, 13(12):2335–2345.

Bilen, H. and Vedaldi, A. (2016). Weakly supervised deep detection networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2846–2854.

Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., and Magli, E. (2019). Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recognition*, 93:134–146.

Bochkovskiy, A. (2017-2023). YOLOv4, YOLOv3 and YOLOv2 for Windows and Linux. `https://github.com/AlexeyAB/darknet`.

Bochkovskiy, A. (2020). Fast-YOLOv4. `https://github.com/AlexeyAB/darknet/blob/master/cfg/yolov4-tiny.cfg`.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint*, arXiv:2004.10934:1–14.

Bookstein, F. L. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585.

Borisyuk, F. et al. (2018). Rosetta: Large scale system for text detection and recognition in images. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 71–79.

Borji, A. (2022). Pros and cons of GAN evaluation measures: New developments. *Computer Vision and Image Understanding*, 215:103329.

Boulahbal, H., Voicila, A., and Comport, A. I. (2021). Are conditional GANs explicitly conditional? In *British Machine Vision Conference (BMVC)*, pages 1–15.

Boureau, Y., Bach, F., LeCun, Y., and Ponce, J. (2010a). Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2559–2566.

Boureau, Y.-L., Ponce, J., and LeCun, Y. (2010b). A theoretical analysis of feature pooling in visual recognition. In *International Conference on Machine Learning*, pages 111–118.

Brownlee, J. (2019). *Generative Adversarial Networks with Python*. `https://machinelearningmastery.com/generative_adversarial_networks/`.

Buslaev, A. et al. (2020). Albumentations: Fast and flexible image augmentations. *Information*, 11(2):125.

CAMEA (2024). UnicamLPR. `https://www.cameatechnology.com/`.

Castro-Zunti, R. D., Yépez, J., and Ko, S.-B. (2020). License plate segmentation and recognition system using deep learning and OpenVINO. *IET Intelligent Transport Systems*, 14(2):119–126.

Chan, L. Y., Zimmer, A., Silva, J. L. d., and Brandmeier, T. (2020). European Union dataset and annotation tool for real time automatic license plate detection and blurring. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848.

Chen, Q. and Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1529.

Chen, S.-L. et al. (2023). End-to-end multi-line license plate recognition with cascaded perception. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 274–289.

Chen, S.-L., Yang, C., Ma, J.-W., Chen, F., and Yin, X.-C. (2020). Simultaneous end-to-end vehicle and license plate detection with multi-branch attention neural network. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3686–3695.

Chen, X., Jin, L., Zhu, Y., Luo, C., and Wang, T. (2022). Text recognition in the wild: A survey. *ACM Computing Surveys*, 54(2):1–35.

Cheng, K., Tahir, R., Eric, L. K., and Li, M. (2020). An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. *Multimedia Tools and Applications*, 79:13725–13752.

Ch'ng, C. K. and Chan, C. S. (2017). Total-Text: A comprehensive dataset for scene text detection and recognition. In *IAPR International Conference on Document Analysis and Recognition*, pages 935–942.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8797.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). StarGAN v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807.

Chollet, F. et al. (2015-2024). Keras. `https://keras.io`.

Chowdhury, P. N. et al. (2020). Graph attention network for detecting license plates in crowded street scenes. *Pattern Recognition Letters*, 140:18–25.

CONTRAN (2007). *RESOLUÇÃO 231 DE 15 DE MARÇO DE 2007 - Estabelece o Sistema de Placas de Identificação de Veículos*. `https://www.gov.br/transportes/pt-br/centrais-de-conteudo/resolucao-231-pdf`. Accessed: 2024-02-21.

Cooijmans, T., Ballas, N., Laurent, C., and Courville, A. C. (2017). Recurrent batch normalization. In *International Conference on Learning Representations (ICLR)*, pages 1–13.

Dai, J., Li, Y., He, K., and Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In *International Conference on Neural Information Processing Systems*, page 379–387.

Dai, S. et al. (2024). Improving small license plate detection with bidirectional vehicle-plate relation. In *International Conference on Multimedia Modeling (MMM)*, pages 253–266.

Das, A. et al. (2023). Urban scene semantic segmentation with low-cost coarse annotation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5967–5976.

Deng, L. and Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387.

Denton, E. L., Chintala, S., szlam, a., and Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 28, pages 1–10.

Ding, H. et al. (2023). Boosting one-stage license plate detector via self-constrained contrastive aggregation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):4204–4216.

Ding, H., Gao, J., Yuan, Y., and Wang, Q. (2024). An end-to-end contrastive license plate detector. *IEEE Transactions on Intelligent Transportation Systems*, 25(1):503–516.

Dlagnekov, L. and Belongie, S. (2005). UCSD/Calit2 car license plate, make and model database. `http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html`.

Dosovitskiy, A. et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, pages 1–22.

Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., Smagt, P. v. d., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766.

Du, S., Ibrahim, M., Shehata, M., and Badawy, W. (2013). Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):311–325.

Dumoulin, V. and Visin, F. (2018). A guide to convolution arithmetic for deep learning. *arXiv preprint*, arXiv:1603.07285:1–31.

Emami, A., Suleman, K., Trischler, A., and Cheung, J. C. K. (2020). An analysis of dataset overlap on Winograd-style tasks. In *International Conference on Computational Linguistics*, pages 5855–5865.

Estevam, V. et al. (2024). Tell me what you see: A zero-shot action recognition method based on natural language descriptions. *Multimedia Tools and Applications*, 83(9):28147–28173.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Fabbrizzi, S., Papadopoulos, S., Ntoutsi, E., and Kompatsiaris, I. (2022). A survey on bias in visual datasets. *Computer Vision and Image Understanding*, 223:103552.

Fan, X. and Zhao, W. (2022). Improving robustness of license plates automatic recognition in natural scenes. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):18845–18854.

Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A. M., Mohamed, S., and Goodfellow, I. (2018). Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In *International Conference on Learning Representations (ICLR)*, pages 1–18.

Feldman, V. and Zhang, C. (2020). What neural networks memorize and why: Discovering the long tail via influence estimation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 2881–2891.

Fernandes, L. S. et al. (2020). A robust automatic license plate recognition system for embedded devices. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 226–239.

Gao, F., Cai, Y., Ge, Y., and Lu, S. (2020a). EDF-LPR: a new encoder-decoder framework for license plate recognition. *IET Intelligent Transport Systems*, 14(8):959–969.

Gao, F., Xu, Y., Ge, Y., Lu, S., and Zhang, Y. (2020b). Property-based shadow detection and removal method for licence plate image. *IET Image Processing*, 14(7):1415–1425.

Gao, Y., Lu, H., Mu, S., and Xu, S. (2023). GroupPlate: Toward multi-category license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5586–5599.

Garcia-Bordils, S. et al. (2022). Out-of-vocabulary challenge report. In *European Conference on Computer Vision (ECCV), TiE: Text in Everything Workshop*, pages 1–17.

Gómez, L., Rusiñol, M., and Karatzas, D. (2018). Cutting sayre's knot: Reading scene text without segmentation. Application to utility meters. In *IAPR International Workshop on Document Analysis Systems*, pages 97–102.

Gonçalves, G. R., da Silva, S. P. G., Menotti, D., and Schwartz, W. R. (2016a). Benchmark for license plate character segmentation. *Journal of Electronic Imaging*, 25(5):053034.

Gonçalves, G. R. et al. (2018). Real-time automatic license plate recognition through deep multi-task networks. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 110–117.

Gonçalves, G. R. et al. (2019). Multi-task learning for low-resolution license plate recognition. In *Iberoamerican Congress on Pattern Recognition (CIARP)*, pages 251–261.

Gonçalves, G. R., Menotti, D., and Schwartz, W. R. (2016b). License plate recognition based on temporal redundancy. In *IEEE International Conference on Intelligent Transportation Systems*, pages 2577–2582.

Gong, Y. et al. (2022). Unified Chinese license plate detection and recognition with high efficiency. *Journal of Visual Communication and Image Representation*, 86:103541.

Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint*, arXiv:1701.00160:1–57.

Goodfellow, I. (2019). Adversarial machine learning. - ICLR 2019 invited talk, `https://www.youtube.com/watch?v=sucqskXRkss`. Accessed: 2024-02-21.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. (2014a). Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *International Conference on Learning Representations (ICLR)*, pages 1–13.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014b). Generative adversarial nets. In *International Conference on Neural Information Processing Systems (NeurIPS)*, page 2672–2680.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pages 369–376.

Graves, A. and Schmidhuber, J. (2005a). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610.

Graves, A. and Schmidhuber, J. (2005b). Framewise phoneme classification with bidirectional LSTM networks. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2047–2052.

Guggenheim, J. A. and Silversmith, J. M. (2000). Confederate license plates at the constitutional crossroads: Vanity plates, special registration organization plates, bumper stickers, viewpoints, vulgarity, and the first amendment. *U. Miami L. Rev.*, 54:563.

Gui, J. et al. (2023). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3313–3332.

Gulrajani, I. et al. (2017). Improved training of Wasserstein GANs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5769–5779.

Han, B.-G., Lee, J. T., Lim, K.-T., and Choi, D.-H. (2020). 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(8):2780.

Harshvardhan, G. M., Gourisaria, M. K., Pandey, M., and Rautaray, S. S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285.

Hasnat, A. and Nakib, A. (2021). Robust license plate signatures matching based on multi-task learning approach. *Neurocomputing*, 440:58–71.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Henry, C., Ahn, S. Y., and Lee, S. (2020). Multinational license plate recognition using generalized character sequence detection. *IEEE Access*, 8:35185–35199.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6629–6640.

Hinton, G., Vinyals, O., and Dean, J. (2014). Distilling the knowledge in a neural network. In *International Conference on Neural Information Processing Systems (NeurIPS) - Deep Learning Workshop*.

Hooker, S., Moorosi, N., Clark, G., Bengio, S., and Denton, E. (2020). Characterising bias in compressed models. *arXiv preprint*, arXiv:2010.03058:1–13. Google Research.

Hort, M., Chen, Z., Zhang, J. M., Harman, M., and Sarro, F. (2023). Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM Journal on Responsible Computing*, pages 1–51.

Hsu, G. S., Ambikapathi, A., Chung, S. L., and Su, C. P. (2017). Robust license plate detection in the wild. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6.

Hsu, G. S., Chen, J. C., and Chung, Y. Z. (2013). Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*, 62(2):552–561.

Hsu, G.-S., Zeng, S.-D., Chiu, C.-W., and Chung, S.-L. (2015). A comparison study on motorcycle license plate detection. In *IEEE International Conference on Multimedia Expo Workshops*, pages 1–6.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

Huang, J. et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3296–3297.

Hwang, S. et al. (2015). Multispectral pedestrian detection: Benchmark dataset and baseline. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1045.

ING Economics (2024). Global car market to hit the speed bumps in 2024. `https://think.ing.com/articles/global-car-market-outlook-hitting-speed-bumps/`. Accessed: 2024-03-20.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456.

Ismail, A., Mehri, M., Sahbani, A., and Amara, N. B. (2021). Performance benchmarking of YOLO architectures for vehicle license plate detection from real-time videos captured by a mobile robot. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 661–668.

Isola, P. et al. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976.

Izidio, D. M. F. et al. (2020). An embedded automatic license plate recognition system using deep learning. *Design Automation for Embedded Systems*, 24:23–43.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2017–2025.

Jaipuria, N., Stevo, K., Zhang, X., Gaopande, M. L., Calle, I., Jain, J., and Murali, V. N. (2022). deepPIC: Deep perceptual image clustering for identifying bias in vision datasets. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4793–4802.

Jia, W. and Xie, M. (2023). An efficient license plate detection approach with deep convolutional neural networks in unconstrained scenarios. *IEEE Access*, 11:85626–85639.

Jiang, Q., Wang, J., Peng, D., Liu, C., and Jin, L. (2023a). Revisiting scene text recognition: A data perspective. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20486–20497.

Jiang, Y., Jiang, F., Luo, H., Lin, H., Yao, J., Liu, J., and Ren, J. (2023b). An efficient and unified recognition method for multiple license plates in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5376–5389.

Jin, X., Tang, R., Liu, L., and Wu, J. (2021). Vehicle license plate recognition for fog-haze environments. *IET Image Processing*, 15(6):1273–1284.

Kaneko, T., Hiramatsu, K., and Kashino, K. (2017). Generative attribute controller with conditional filtered generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7006–7015.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116.

Ke, X., Zeng, G., and Guo, W. (2023). An ultra-fast automatic license plate recognition approach for unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5172–5185.

Kessentini, Y., Besbes, M. D., Ammar, S., and Chabbouh, A. (2019). A two-stage deep neural network for multi-norm license plate detection and recognition. *Expert Systems with Applications*, 136:159–170.

Khan, K. et al. (2021). Performance enhancement method for multiple license plate recognition in challenging environments. *EURASIP Journal on Image and Video Processing*, 2021(1):30.

Khosla, A., Zhou, T., Malisiewicz, T., Efros, A. A., and Torralba, A. (2012). Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, pages 158–171.

Kim, D., Kim, J., and Park, E. (2024). AFA-Net: Adaptive feature attention network in image deblurring and super-resolution for improving license plate recognition. *Computer Vision and Image Understanding*, 238:103879.

Kim, T. et al. (2017). Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning (ICML)*, volume 70, pages 1857–1865.

Kim, T.-G., Yun, B.-J., Kim, T.-H., Lee, J.-Y., Park, K.-H., Jeong, Y., and Kim, H. D. (2021). Recognition of vehicle license plates based on image processing. *Applied Sciences*, 11(14):6292.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, pages 1–15.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, pages 1–14.

Kong, X. et al. (2021). A federated learning-based license plate recognition scheme for 5G-enabled internet of vehicles. *IEEE Transactions on Industrial Informatics*, 17(12):8523–8530.

Krizhevsky, A. et al. (2012). ImageNet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1097–1105.

Kurpiel, F. D., Minetto, R., and Nassu, B. T. (2017). Convolutional neural networks for license plate detection in images. In *IEEE International Conference on Image Processing (ICIP)*, pages 3395–3399.

Laroca, R., Araujo, A. B., Zanlorensi, L. A., De Almeida, E. C., and Menotti, D. (2021a). Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach. *IEEE Access*, 9:67569–67584.

Laroca, R., Barroso, V., Diniz, M. A., Gonçalves, G. R., Schwartz, W. R., and Menotti, D. (2019). Convolutional neural networks for automatic meter reading. *Journal of Electronic Imaging*, 28(1):013023.

Laroca, R., Cardoso, E. V., Lucio, D. R., Estevam, V., and Menotti, D. (2022a). On the cross-dataset generalization in license plate recognition. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 166–178.

Laroca, R., Estevam, V., Britto Jr., A. S., Minetto, R., and Menotti, D. (2023a). Do we train on test data? The impact of near-duplicates on license plate recognition. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Laroca, R., Santos, M., Estevam, V., Luz, E., and Menotti, D. (2022b). A first look at dataset bias in license plate recognition. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 234–239.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., and Menotti, D. (2018). A robust real-time automatic license plate recognition based on the YOLO detector. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–10.

Laroca, R., Zanlorensi, L. A., Estevam, V., Minetto, R., and Menotti, D. (2023b). Leveraging model fusion for improved license plate recognition. In *Iberoamerican Congress on Pattern Recognition (CIARP)*, pages 60–75.

Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., and Menotti, D. (2021b). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intelligent Transport Systems*, 15(4):483–503.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, C. and Osindero, S. (2016). Recursive recurrent nets with attention modeling for OCR in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2231–2239.

Lee, H.-Y. et al. (2020). DRIT++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128:2402–2417.

Lee, Y., Jeon, J., Ko, Y., Jeon, M., and Pedrycz, W. (2022). License plate detection via information maximization. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14908–14921.

Lee, Y., Lee, J., Ahn, H., and Jeon, M. (2019). SNIDER: Single noisy image denoising and rectification for improving license plate recognition. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1017–1026.

Lehner, A. et al. (2024). 3D adversarial augmentations for robust out-of-domain predictions. *International Journal of Computer Vision*, 132(3):931–963.

Li, H., Wang, P., and Shen, C. (2019). Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1126–1136.

Li, H., Wang, P., You, M., and Shen, C. (2018). Reading car license plates using deep neural networks. *Image and Vision Computing*, 72:14–23.

Li, Z., Wang, F., Taleb, H., Yuan, C., Qin, X., Wu, H., Zhao, X., and Zhang, L. (2020). License plate detection and recognition technology for complex real scenarios. In *International Conference on Intelligent Computing (ICIC)*, pages 241–256.

Liang, J., Chen, G., Wang, Y., and Qin, H. (2022). EGSANet: edge–guided sparse attention network for improving license plate detection in the wild. *Applied Intelligence*, 52(4):4458–4472.

Liang, M. and Hu, X. (2015). Recurrent convolutional neural network for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3367–3375.

Liao, T., Taori, R., Raji, I. D., and Schmidt, L. (2021). Are we learning yet? a meta review of evaluation failures across machine learning. In *International Conference on Neural Information Processing Systems (NeurIPS). Datasets and Benchmarks Track*, pages 1–10.

Lin, M., Chen, Q., and Yan, S. (2014a). Network in network. In *International Conference on Learning Representations (ICLR)*, pages 1–10.

Lin, T.-Y. et al. (2014b). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755.

Liu, C. and Chang, F. (2019). Hybrid cascade structure for license plate detection in large visual surveillance scenes. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2122–2135.

Liu, J. et al. (2020). Boosting semantic human matting with coarse annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8560–8569.

Liu, Q., Chen, S.-L., Li, Z.-J., Yang, C., Chen, F., and Yin, X.-C. (2021). Fast recognition for multidirectional and multi-type license plates with 2D spatial attention. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 125–139.

Liu, Q. et al. (2024a). Improving multi-type license plate recognition via learning globally and contrastively. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11. Early Access.

Liu, W., Chen, C., Kwan-Yee K. Wong, Z. S., and Han, J. (2016). STAR-Net: A spatial attention residue network for scene text recognition. In *British Machine Vision Conference (BMVC)*, pages 1–13.

Liu, Y.-Y., Liu, Q., Chen, S.-L., Chen, F., and Yin, X.-C. (2024b). Irregular license plate recognition via global information integration. In *International Conference on Multimedia Modeling*, pages 325–339.

Lu, X., Yuan, Y., and Wang, Q. (2021). AWFA-LPD: Adaptive weight feature aggregation for multi-frame license plate detection. In *International Conference on Multimedia Retrieval (ICMR)*, pages 476–480.

Lubna, Mufti, N., and Shah, S. A. A. (2021). Automatic number plate Recognition: A detailed survey of relevant algorithms. *Sensors*, 21(9):3028.

Lucio, D. R. et al. (2019). Simultaneous iris and periocular region detection using coarse annotations. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 178–185.

Lučić, M., Tschannen, M., Ritter, M., Zhai, X., Bachem, O., and Gelly, S. (2019). High-fidelity image generation with fewer labels. In *International Conf. on Machine Learning (ICML)*, pages 4183–4192.

Ma, Z., Hong, X., Wei, X., Qiu, Y., and Gong, Y. (2021). Towards a universal model for cross-dataset crowd counting. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3185–3194.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, pages 1–6.

Maier, A., Moussa, D., Spruck, A., Seiler, J., and Riess, C. (2022). Reliability scoring for the recognition of degraded license plates. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8.

Masood, S. Z., Shu, G., Dehghan, A., and Ortiz, E. G. (2017). License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint*, arXiv:1703.07330.

McLaughlin, N. et al. (2015). Data-augmentation for reducing dataset bias in person re-identification. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Mecocci, A. and Tommaso, C. (2006). Generative models for license plate recognition by using a limited number of training samples. In *International Conference on Image Processing*, pages 2769–2772.

Mendes Júnior, P. R., Neves, J. M. R., Tavares, A. I., and Menotti, D. (2011). Towards an automatic vehicle access control system: License plate location. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2916–2921.

Meng, A., Yang, W., Xu, Z., Huang, H., Huang, L., and Ying, C. (2018). A robust and efficient method for license plate recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 1713–1718.

MERCOSUR (2014). *Res GMC 33-14 - Patente y Sistema de Consultas Sobre Vehículos del MERCOSUR*. https://www.mercosur.int/documento/res-gmc-33-14/. Accessed: 2024-02-21.

MERCOSUR (2024). MERCOSUR Countries. https://www.mercosur.int/en/about-mercosur/mercosur-countries/. Accessed: 2024-02-21.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint*, arXiv:1411.1784:1–7.

Misra, D. (2020). Mish: A self regularized non-monotonic activation function. In *British Machine Vision Conference (BMVC)*, pages 1–14.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, pages 1–26.

Miyato, T. and Koyama, M. (2018). cGANs with projection discriminator. *International Conference on Learning Representations (ICLR)*, pages 1–21.

Mokayed, H. et al. (2021). A new DCT-PCM method for license plate number detection in drone images. *Pattern Recognition Letters*, 148:45–53.

Moussa, D. et al. (2022). Forensic license plate recognition with compression-informed transformers. In *IEEE International Conference on Image Processing (ICIP)*, pages 406–410.

Nader, A. and Azar, D. (2020). Searching for activation functions using a self-adaptive evolutionary algorithm. In *Genetic and Evolutionary Computation Conference Companion (GECCO)*, page 145–146.

Nascimento, V. et al. (2022). Combining attention module and pixel shuffle for license plate super-resolution. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 228–233.

Nascimento, V. et al. (2023). Super-resolution of license plate images using attention modules and sub-pixel convolution layers. *Computers & Graphics*, 113:69–76.

Nguyen, H. (2022). An efficient license plate detection approach using lightweight deep convolutional neural networks. *Advances in Multimedia*, 2022:8852142.

Nguyen-Phuoc, D. Q., Truong, T. M., Nguyen, M. H., Pham, H.-G., Li, Z.-C., and Oviedo-Trespalacios, O. (2024). What factors influence the intention to use electric motorcycles in motorcycle-dominated countries? an empirical study in Vietnam. *Transport Policy*, 146:193–204.

Odena, A., Buckman, J., Olsson, C., Brown, T., Olah, C., Raffel, C., and Goodfellow, I. (2018). Is generator conditioning causally related to GAN performance? In *International Conference on Machine Learning (ICML)*, pages 3849–3858.

Oliveira, I. O. et al. (2021). Vehicle-Rear: A new dataset to explore feature fusion for vehicle identification using convolutional neural networks. *IEEE Access*, 9:101065–101077.

OpenALPR (2016). OpenALPR datasets. https://github.com/openalpr/benchmarks/tree/master/endtoend/. Accessed: 2024-02-21.

OpenALPR (2022). OpenALPR Cloud API (2022 release). http://www.openalpr.com/.

OpenALPR (2023). OpenALPR Cloud API (2023 release). http://www.openalpr.com/.

OpenALPR (2024). OpenALPR Cloud API. http://www.openalpr.com/.

Oussidi, A. and Elhassouny, A. (2018). Deep generative models: Survey. In *International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–8.

Padilla, R., Netto, S. L., and da Silva, E. A. B. (2020). A survey on performance metrics for object-detection algorithms. In *International Conference on Systems, Signals and Image Processing*, pages 237–242.

Pan, X., Li, S., Li, R., and Sun, N. (2022). A hybrid deep learning algorithm for the license plate detection and recognition in vehicle-to-vehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):23447–23458.

Panahi, R. and Gholampour, I. (2017). Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):767–779.

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341.

Pattanaik, A. and Balabantaray, R. C. (2023). Enhancement of license plate recognition performance using Xception with Mish activation function. *Multimedia Tools and Applications*, 82(11):16793–16815.

Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. In *ACM SIGGRAPH*, page 313–318.

Pham, T.-A. (2023). Effective deep neural networks for license plate detection and recognition. *The Visual Computer*, 39(3):927–941.

Polikar, R. (2012). *Ensemble Learning*, pages 1–34. Springer.

Ponce, J., Berg, T. L., Everingham, M., Forsyth, D. A., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B. C., Torralba, A., Williams, C. K. I., Zhang, J., and Zisserman, A. (2006). Dataset issues in object recognition. In *Toward Category-Level Object Recognition*, pages 29–48. Springer.

Ponti, M. A., Ribeiro, L. S. F., Nazare, T. S., Bui, T., and Collomosse, J. (2017). Everything you wanted to know about deep learning for computer vision but were afraid to ask. In *Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 17–41.

Poursaeed, O. et al. (2021). Robustness and generalization via generative adversarial training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15691–15700.

Powers, D. M. W. (2015). What the F-measure doesn't measure: Features, flaws, fallacies and fixes. *arXiv preprint*, arXiv:1503.06410.

Presidência da República (1997). *LEI Nº 9.503, DE 23 DE SETEMBRO DE 1997 - Código de Trânsito Brasileiro*. `http://www.planalto.gov.br/ccivil_03/leis/l9503compilado.htm`.

Qiao, L., Chen, Y., Cheng, Z., Xu, Y., Niu, Y., Pu, S., and Wu, F. (2021). MANGO: A mask attention guided one-stage scene text spotter. *AAAI Conference on Artificial Intelligence*, 35(3):2467–2476.

Qin, S. and Liu, S. (2020). Efficient and unified license plate recognition via lightweight deep neural network. *IET Image Processing*, 14(16):4102–4109.

Qin, S. and Liu, S. (2022). Towards end-to-end car license plate location and recognition in unconstrained scenarios. *Neural Computing and Applications*, 34:21551–21566.

Radford, A. et al. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, pages 1–16.

Ramachandran, P., Zoph, B., and Le, Q. V. (2018). Searching for activation functions. In *International Conference on Learning Representations (ICLR) – Workshop Session*, pages 1–13.

Rao, Z. et al. (2024). License plate recognition system in unconstrained scenes via a new image correction scheme and improved CRNN. *Expert Systems with Applications*, 243:122878.

Redmon, J. (2018). Computers can see. Now what? - TEDxGateway, `https://www.youtube.com/watch?v=XS2UWYuh5u0`. Accessed: 2024-02-21.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Redmon, J. and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint*, arXiv:1804.02767:1–6.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International Conference on Machine Learning (ICML)*, pages 1060–1069.

Ren, S. et al. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Ribeiro, V., Greati, V., Bezerra, A., Silvano, G., Silva, I., Endo, P. T., and Lynn, T. (2019). Brazilian Mercosur license plate detection: a deep learning approach relying on synthetic imagery. In *Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–8.

Rocha, C. V. M. et al. (2022). A chatbot solution for self-reading energy consumption via chatting applications. *Journal of Control, Automation and Electrical Systems*, 33(1):229–240.

RodoSol (2024). *Concessionária Rodovia do Sol S/A*. `https://web.archive.org/web/20110101142314/https://www.rodosol.com.br/blog/conheca-a-rodosol-2`.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2234–2242.

Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 901–909.

Salomon, G. et al. (2020). Deep learning for image-based automatic dial meter reading: Dataset and baselines. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Sandler, M. et al. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520.

Schirrmacher, F. et al. (2023). Benchmarking probabilistic deep learning methods for license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*, 24(9):9203–9216.

Seibel, H., Goldenstein, S., and Rocha, A. (2017). Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos. *IEEE Access*, 5:20020–20035.

Selmi, Z., Halima, M. B., Pal, U., and Alimi, M. A. (2020). DELP-DAR system for license plate detection and recognition. *Pattern Recognition Letters*, 129:213–223.

Selvaraju, R. R. et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.

Senatran (2024). Frota de Veiculos - 2024. `https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-Senatran/frota-de-veiculos-2024`. Accessed: 2024-03-14.

Serajeh, R. (2016). Two lines Iranian license plate detection and recognition using subspace learning. In *International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–6.

Shaham, T. R., Gharbi, M., Zhang, R., Shechtman, E., and Michaeli, T. (2021). Spatially-adaptive pixelwise networks for fast image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14877–14886.

Shane Barratt, R. S. (2018). A note on the inception score. *arXiv preprint*, arXiv:1801.01973:1–9.

Shashirangana, J. et al. (2022). License plate recognition using neural architecture search for edge devices. *International Journal of Intelligent Systems*, 37(12):10211–10248.

Shashirangana, J., Padmasiri, H., Meedeniya, D., and Perera, C. (2021). Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, 9:11203–11225.

Shi, B., Bai, X., and Yao, C. (2017). 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.

Shi, B. et al. (2019). ASTER: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2035–2048.

Shi, B., Wang, X., Lyu, P., Yao, C., and Bai, X. (2016). Robust scene text recognition with automatic rectification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4168–4176.

Shmelkov, K., Schmid, C., and Alahari, K. (2018). How good is my GAN? In *European Conference on Computer Vision (ECCV)*, pages 218–234.

Shu, C. et al. (2020). YM-NET: A new network structure for license plate detection in complex scenarios. In *International Conference on Aviation Safety and Information Technology*, page 600–605.

Shvai, N., Hasnat, A., and Nakib, A. (2023). Multiple auxiliary classifiers GAN for controllable image generation: Application to license plate recognition. *IET Intelligent Transport Systems*, 17(1):243–254.

Sighthound (2022). Sighthound ALPR+ (2022 version). `www.sighthound.com/products/alpr`.

Sighthound (2023). Sighthound ALPR+ (2023 version). `www.sighthound.com/products/alpr`.

Sighthound (2024). Sighthound ALPR+. `www.sighthound.com/products/alpr`.

Silva, S. M. and Jung, C. R. (2017). Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In *Conference on Graphics, Patterns and Images*, pages 55–62.

Silva, S. M. and Jung, C. R. (2018). License plate detection and recognition in unconstrained scenarios. In *European Conference on Computer Vision (ECCV)*, pages 593–609.

Silva, S. M. and Jung, C. R. (2020). Real-time license plate detection and recognition using deep convolutional neural networks. *J. of Visual Communication and Image Representation*, page 102773.

Silva, S. M. and Jung, C. R. (2022). A flexible approach for automatic license plate recognition in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5693–5703.

Silvano, G., Ribeiro, V., Greati, V., Bezerra, A., Silva, I., Endo, P. T., and Lynn, T. (2021). Synthetic image generation for training deep learning-based automated license plate recognition systems on the Brazilian Mercosur standard. *Design Automation for Embedded Systems*, 25(2):113–133.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, pages 1–12.

Špaňhel, J. et al. (2018). Geometric alignment by deep learning for recognition of challenging license plates. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 3524–3529.

Špaňhel, J., Sochor, J., Juránek, R., Herout, A., Maršík, L., and Zemčík, P. (2017). Holistic recognition of low quality license plates by CNN using track annotated data. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6.

Srebrić, V. (2003). EnglishLP database. `https://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip`.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Statista (2024). Global automotive market share in 2023, by brand. `https://www.statista.com/statistics/316786/global-market-share-of-the-leading-automakers/`.

Sun, M., Zhou, F., Yang, C., and Yin, X. (2019). Image generation framework for unbalanced license plate data set. In *International Conference on Data Mining Workshops (ICDMW)*, pages 883–889.

Szegedy, C. et al. (2015). Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

Szegedy, C. et al. (2016). Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Tang, H., Liu, H., Xu, D., Torr, P. H. S., and Sebe, N. (2021). AttentionGAN: Unpaired image-to-image translation using attention-guided generative adversarial networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–16.

Tejani, S. (2016). Machines that can see: Convolutional neural networks. `https://shafeentejani.github.io/2016-12-20/convolutional-neural-nets/`. Accessed: 2024-02-21.

Theis, L., van den Oord, A., and Bethge, M. (2016). A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR)*, pages 1–10.

Thome, N., Vacavant, A., Robinault, L., and Miguet, S. (2011). A cognitive and video-based approach for multinational license plate recognition. *Machine Vision and Applications*, 22(2):389–407.

Tian, M. et al. (2018). Eliminating background-bias for robust person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5794–5803.

Tommasi, T., Patricia, N., Caputo, B., and Tuytelaars, T. (2017). A deeper look at dataset bias. In *Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer.

Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528.

Touvron, H. et al. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pages 10347–10357.

Trinh, L., Pham, P., Trinh, H., Bach, N., Nguyen, D., Nguyen, G., and Nguyen, H. (2023). PP4AV: A benchmarking dataset for privacy-preserving autonomous driving. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1206–1215.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *International Conference on Neural Information Processing Systems (NeurIPS)*, page 6000–6010.

Vašek, V. et al. (2018). License plate recognition and super-resolution from low-resolution videos by convolutional neural networks. In *British Machine Vision Conference (BMVC)*, pages 1–12.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. *International Conference on Learning Representations (ICLR)*, pages 1–12.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

Wachinger, C., Rieckmann, A., and Pölsterl, S. (2021). Detect and correct bias in multi-site neuroimaging datasets. *Medical Image Analysis*, 67:101879.

Wan, Z., Zhang, J., Zhang, L., Luo, J., and Yao, C. (2020). On vocabulary reliance in scene text recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11422–11431.

Wang, B. et al. (2022a). Character segmentation and recognition of variable-length license plates using ROI detection and broad learning system. *Remote Sensing*, 14(7):1560.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2021a). Scaled-YOLOv4: Scaling cross stage partial network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13029–13038.

Wang, J. and Hu, X. (2017). Gated recurrent convolution neural network for OCR. In *International Conference on Neural Information Processing Systems (NeurIPS)*, page 334–343.

Wang, J., Huang, H., Qian, X., Cao, J., and Dai, Y. (2018a). Sequence recognition of chinese license plates. *Neurocomputing*, 317:149–158.

Wang, T., Wang, W., Li, C., and Tang, J. (2022b). Efficient license plate recognition via parallel position-aware attention. In *Pattern Recognition and Computer Vision*, pages 346–360.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018b). High-resolution image synthesis and semantic manipulation with conditional GANs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807.

Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., and Shao, S. (2019). Shape robust text detection with progressive scale expansion network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9328–9337.

Wang, X., You, M., and Shen, C. (2017). Adversarial generation of training examples for vehicle license plate recognition. *arXiv preprint*, arXiv:1503.06410.

Wang, Y. et al. (2022c). Rethinking and designing a high-performing automatic license plate recognition approach. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8868–8880.

Wang, Z., She, Q., and Ward, T. E. (2021b). Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys*, 54(2):1–38.

Weber, M. (1999). Caltech Cars dataset. `https://data.caltech.edu/records/20084`.

Weihong, W. and Jiaoyang, T. (2020). Research on license plate recognition algorithms based on deep learning in complex environment. *IEEE Access*, 8:91661–91675.

Wojna, Z., Gorban, A. N., Lee, D.-S., Murphy, K., Yu, Q., Li, Y., and Ibarz, J. (2017). Attention-based extraction of structured information from street view imagery. In *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 844–850.

Wu, C., Xu, S., Song, G., and Zhang, S. (2018). How many labeled license plates are needed? In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 334–346.

Wu, G. et al. (2022). CarveNet: a channel-wise attention-based network for irregular scene text recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 25(3):177–186.

Wu, S., Zhai, W., and Cao, Y. (2019). PixTextGAN: structure aware text image synthesis for license plate recognition. *IET Image Processing*, 13(14):2744–2752.

Xiang, H., Zhao, Y., Yuan, Y., Zhang, G., and Hu, X. (2019). Lightweight fully convolutional network for license plate detection. *Optik*, 178:1185–1194.

Xie, L., Ahmad, T., Jin, L., Liu, Y., and Zhang, S. (2018). A new CNN-based method for multi-directional car license plate detection. *IEEE Transactions on Intelligent Transportation Systems*, 19(2):507–517.

Xu, H. et al. (2021). 2D license plate recognition based on automatic perspective rectification. In *International Conference on Pattern Recognition (ICPR)*, pages 202–208.

Xu, H., Zhou, X.-D., Li, Z., Liu, L., Li, C., and Shi, Y. (2022). EILPR: Toward end-to-end irregular license plate recognition based on automatic perspective alignment. *IEEE Transactions on Intelligent*

Transportation Systems, 23(3):2586–2595.

Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C., and Huang, L. (2018). Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *European Conference on Computer Vision (ECCV)*, pages 261–277.

Yang, X., Zhang, B., and Lien, K.-C. (2023). SATPlate: A Germany license plate detection dataset and baselines. In *IEEE International Conference on Image Processing (ICIP)*, pages 3329–3333.

Yang, Y., Li, D., and Duan, Z. (2018). Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features. *IET Intelligent Transport Systems*, 12(3):213–219.

Yi, Z., Zhang, H., Tan, P., and Gong, M. (2017). DualGAN: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2868–2876.

Yonetsu, S., Iwamoto, Y., and Chen, Y. W. (2019). Two-stage YOLOv2 for accurate license-plate detection in complex scenes. In *IEEE International Conference on Consumer Electronics*, pages 1–4.

Yoo, H. and Jun, K. (2021). Deep corner prediction to rectify tilted license plate images. *Multimedia Systems*, 27(4):779–786.

Yu, C. et al. (2018). BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 334–349.

Yuan, Y., Zou, W., Zhao, Y., Wang, X., Hu, X., and Komodakis, N. (2017). A robust and efficient approach to license plate detection. *IEEE Transactions on Image Processing*, 26(3):1102–1114.

Yuniaristanto, Sutopo, W., Hisjam, M., and Wicaksono, H. (2024). Exploring the determinants of intention to purchase electric motorcycles: The role of national culture in the UTAUT. *Transportation Research Part F: Traffic Psychology and Behaviour*, 100:475–492.

Zeni, L. F. and Jung, C. (2020). Weakly supervised character detection for license plate recognition. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 218–225.

Zhang, C., Wang, Q., and Li, X. (2020a). EQ-LPR: Efficient quality-aware license plate recognition. In *IEEE International Conference on Image Processing (ICIP)*, pages 653–657.

Zhang, C., Wang, Q., and Li, X. (2020b). IQ-STAN: Image quality guided spatio-temporal attention network for license plate recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2268–2272.

Zhang, C., Wang, Q., and Li, X. (2021a). V-LPDR: Towards a unified framework for license plate detection, tracking, and recognition in real-world traffic videos. *Neurocomputing*, 449:189–206.

Zhang, H. et al. (2021b). Cross-modal contrastive learning for text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 833–842.

Zhang, J., Li, W., Ogunbona, P., and Xu, D. (2019a). Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Computing Surveys*, 52(1):7.

Zhang, L. et al. (2021c). A robust attentional framework for license plate recognition in the wild. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):6967–6976.

Zhang, L., Wang, P., Dang, F., and Zhang, S. (2019b). A simple and robust attentional encoder-decoder model for license plate recognition. In *Pattern Recognition and Computer Vision*, pages 295–307.

Zhang, M., Liu, W., and Ma, H. (2018a). Joint license plate super-resolution and recognition in one multi-task GAN framework. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1443–1447.

Zhang, S. et al. (2019c). An improved vehicle-license plate recognition based on color clues and coding rules. In *International Conference on Image, Vision and Computing (ICIVC)*, pages 504–508.

Zhang, S., Tang, G., Liu, Y., and Mao, H. (2020c). Robust license plate recognition with shared adversarial training network. *IEEE Access*, 8:697–705.

Zhang, X., Gu, N., Ye, H., and Lin, C. (2018b). Vehicle license plate detection and recognition using deep neural networks and generative adversarial networks. *Journal of Electronic Imaging*, 27(4):043056.

Zhang, Y., Wang, Z., and Zhuang, J. (2021d). Efficient license plate recognition via holistic position attention. In *AAAI Conference on Artificial Intelligence*, pages 3438–3446.

Zhao, L., Liu, T., Peng, X., and Metaxas, D. (2020a). Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 14435–14447.

Zhao, Y., Wu, R., and Dong, H. (2020b). Unpaired image-to-image translation using adversarial consistency loss. In *European Conference on Computer Vision (ECCV)*, pages 800–815.

Zheng, C., Cham, T.-J., and Cai, J. (2021). The spatially-correlative loss for various image translation tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16402–16412.

Zhou, W., Li, H., Lu, Y., and Tian, Q. (2012). Principal visual word discovery for automatic license plate detection. *IEEE Transactions on Image Processing*, 21(9):4269–4279.

Zhou, X., Cheng, Y., Jiang, L., Ning, B., and Wang, Y. (2023). FAFEnet: A fast and accurate model for automatic license plate detection and recognition. *IET Image Processing*, 17(3):807–818.

Zhou, X. et al. (2017). EAST: An efficient and accurate scene text detector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2651.

Zhou, X. et al. (2021). CoCosNet v2: Full-resolution correspondence learning for image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11460–11470.

Zhu, J.-Y. et al. (2017a). Toward multimodal image-to-image translation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 465–476.

Zhu, J.-Y. et al. (2017b). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251.

Zhuang, J., Hou, S., Wang, Z., and Zha, Z.-J. (2018). Towards human-level license plate recognition. In *European Conference on Computer Vision (ECCV)*, pages 314–329.

Zibani, R., Sebbak, F., Boudaren, M. E. Y., Mataoui, M., Hadj Aissa, R., and Benaissa, Y. A. (2024). Multi-attribute fusion-based approach for Algerian automatic license plate recognition. *Multimedia Tools and Applications*, 83(10):30233–30259.

Zou, Y., Zhang, Y., Yan, J., Jiang, X., Huang, T., Fan, H., and Cui, Z. (2020). A robust license plate recognition model based on Bi-LSTM. *IEEE Access*, 8:211630–211641.

Zou, Y., Zhang, Y., Yan, J., Jiang, X., Huang, T., Fan, H., and Cui, Z. (2022). License plate detection and recognition based on YOLOv3 and ILPRNET. *Signal, Image and Video Processing*, 16(2):473–480.