UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

CLEITON SOUZA LIMA

Using a super resolution network and general-purpose optical character recognition for license plate recognition.

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering

Advisor: Prof. Dr. Cláudio Jung

Porto Alegre
October 2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos André Bulhões
Vice-Reitora: Profª. Patricia Pranke
Pró-Reitora de Graduação: Profª. Cíntia Inês Boll
Diretora do Instituto de Informática: Profª. Carla Maria Dal Sasso Freitas
Diretora da Escola de Engenharia: Prof. Carla Schwengber Ten Caten
Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages
Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro
Bibliotecária-Chefe da Escola de Engenharia: Prof. Rosane Beatriz Allegretti Borges

“Persistence is the shortest path to success.”

— CHARLES CHAPLIN

DEDICATION

First of all, I would like to dedicate this thesis to my family. Thanks to my mother, Zeli dos Santos Souza, and my father, Jair de Jesus Lima, I was able to achieve all my goals and, without their support, I would not be here. This accomplishment is also yours! Thank you to my sister, Michele, not because she helped me over nights of study or was comprehensive with the thousands hours I spent isolated in my room, but because she was always smiling and trying to make me laugh even when I was in a bad mood or worried about college. Thank you to all my friends and colleagues who shared with me the efforts and challenges of my journey and stood by me in the most diverse situations. It was not an easy period in my life and your support was definitely fundamental. Finally, I thank Professor Dr. Claudio Rosito Jung for the many hours he dedicated to this work and for all the conversations and ideas he shared with me.

ABSTRACT

The extraction and identification of license plates from images have numerous applications and can be used to automate and improve various processes in our society. By using artificial intelligence, we can extract information without human interaction, reducing errors, saving resources, and increasing the number of applicable use cases. In recent years, several applications and studies have been conducted in this area, and a common problem is the low resolution of images. In this monograph, we describe the implementation of a license plate identification pipeline using neural networks for different purposes and address this particular problem by introducing a Text Super-Resolution Network (TSRN). Besides that, we analyze the results by combining different optical character recognition (OCR) modules aiming to increase the system accuracy and robustness. Our experimental results indicate that pre-trained OCR approaches perform very poorly for recognizing license plates, but fine-tuning them with license plate images strongly improves the results. TSRN performed satisfactorily and produced well-defined high-resolution image, but the overall OCR accuracy presented a marginal gain. We believe that the low-resolution problem we were trying to solve with this network was not the critical one in our test dataset.

Keywords: Automatic License Plate Recognition. Super Resolution Neural Network. Low Resolution Images. Neural Network. Optical Character Recognition. Machine Learning.

Usando uma rede de super resolução e reconhecimento ótico de caracteres de propósito geral para reconhecimento de placas veiculares.

RESUMO

Extração e identificação de placas veiculares a partir de imagens têm diversas aplicações e podem ser utilizadas para automatizar e melhorar inúmeros processos na nossa sociedade. Além disso, usando inteligência artificial, é possível extrair informações sem interação humana, o que pode reduzir falhas, economizar recursos e aumentar o número de casos de uso aplicáveis. Nos últimos anos, diversas aplicações e estudos foram conduzidos nessa área e um problema frequente está relacionado às imagens de baixa resolução. Dessa forma, neste trabalho, iremos descrever a implementação de um fluxo de execução para identificar placas veiculares usando redes neurais de diferentes propósitos e endereçar esse problema em particular introduzindo uma Rede de Super Resolução de Texto. Além disso, analisamos os resultados obtidos combinando diferentes modelos de reconhecimento óptico de caracteres visando aumentar a acurácia e robusteza do sistema. Nossos resultados experimentais indicam que as abordagens de OCR pré-treinadas têm um desempenho muito ruim no reconhecimento de placas, mas realizar o *fine-tuning* com imagens de placas melhora consideravelmente os resultados. A técnica TSRN teve um desempenho satisfatório e produziu imagens de alta resolução bem definidas, mas a precisão geral do OCR apresentou um ganho marginal. Acreditamos que o problema de baixa resolução que estávamos tentando resolver com essa rede não era o crítico em nosso conjunto de dados de teste.

Palavras-chave: Reconhecimento de placas veiculares automático, rede neural de super resolução, imagens de baixa resolução, rede neural, reconhecimento de caracteres óptico, aprendizagem de máquina .

LIST OF ABBREVIATIONS AND ACRONYMS

TSRN	Text Super Resolution Network
LP	License Plate
LR	Low Resolution
HR	High Resolution
SR	Super Resolution
DP	Deep Learning
OCR	Optical Character Recognition
IBGE	Brazilian Institute of Geography and Statistics
ALPR	Automatic License Plate Recognition
LPD	License Plate Detection
WPOD	Warped Planar Object Detection Network
RNN	Recurrent Neural Network
CRNN	Convolutional Recurrent Neural Network
ASRN	Attention-based Sequence Recognition Network
LSTM	Long Short-Term Memory
STN	Spatial Transform Network
TPS	Thin Plate Splines
MSE	Mean Squared Error

LIST OF FIGURES

LIST OF TABLES

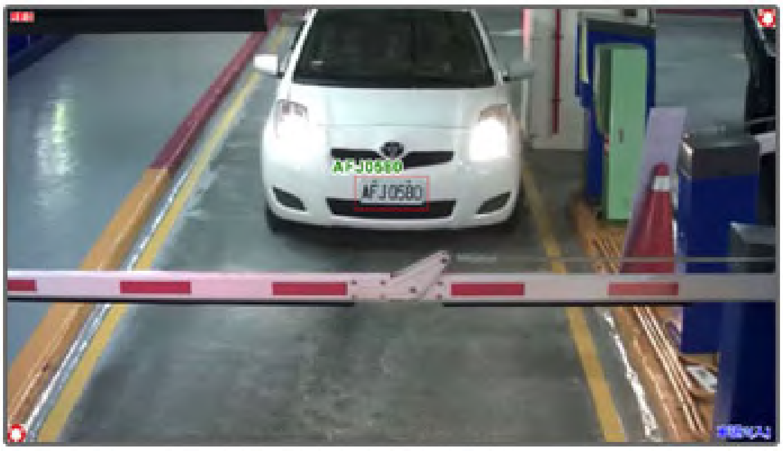
CONTENTS

1 INTRODUCTION

The license plate (LP) is the most important information about a vehicle from the point of view of security and authorities since this unique sequence of characters is able to identify a particular car, motorcycle, or other vehicle traveling on the roads of a country or region. In Brazil, for example, there were more than 111 million registered vehicles in 2021, according to the Brazilian Institute of Geography and Statistics (IBGE) [1]. The unique LP number can be used for different applications: for security reasons, surveillance cameras can be used to track the path of a vehicle; in parking lots, the license plate number can be used to calculate the number of hours a car has spent there; in traffic, LPs are used to check for violations of the law.

Finding a way to automate the identification of LPs can improve efficiency and reduce costs in several areas. However, most automatic license plate recognition (ALPR) solutions on the market require environment specifications in terms of camera placement or design, which reduces the number of applicable use cases. As an example, Figure 1.1 shows a parking entrance control solution [2] that requires the camera position to be set to capture frontal vehicle images in order to correctly detect and recognize the LP. According to the manufacturer, their solution achieves high accuracy, and in this case, taking control over the scene and the position of the object is not a problem. However, it is not suitable for all cases where ALPR could be used.

Figure 1.1 – Example of ALPR system available in the market with restrictions related to the object position for LP recognition.



In this way, solutions that accept images without a predefined camera setup are

[1] IBGE - Licensed vehicles in Brazil (2021): https://cidades.ibge.gov.br/brasil/pesquisa/22/28120

[2] Ticketless ANPR solution from Parkxper company: https://parkxper.com/lpr-e.html

very flexible to adapt to common situations, but the detected license plate may contain artifacts such as blur, saturation, light variations, or low resolution. In such cases, the final task of recognizing the LP string can be compromised.

1.1 Goals

This work tackles a particular type of artifact expected to arise in generic capture setups: low-resolution license plates, which arise when the vehicle is sufficiently distant from the camera. We will explore the combination of a Text Super-Resolution Network (TSRN) with Optical Character Recognition (OCR) networks aiming to improve the recognition rate of low-resolution license plates. As a starting point, we will use the TSRN proposed in Wang et al. (2020b) to deal with low-resolution images, and evaluate different training strategies that either aim to maximize the quality of the super-resolved image or integrate an OCR module and maximize its recognition rate.

2 RELATED WORK

Automatic License Plate Recognition (ALPR) systems are typically developed as a pipeline in which each step or module is responsible for a specific task. In an overview of the ALPR system defined in this work, we can see three main fronts: the license plate detection module, which is responsible for license plate identification and extraction; the license plate recognition module, which is responsible for converting the text image into characters; and, as one of the proposals of this work, the super-resolution module, that will handle low-resolution images. In the next sections, we will introduce in detail each of these modules used to build the ALPR pipeline used in this work and the publications where the modules were originally proposed.

2.1 License Plate Detection

Deep learning techniques have improved various image processing tasks such as object detection and optical character recognition. Over the years, several CNN architectures have been proposed to handle license plate detection (LPD) typically based on generic object detectors, either in one-stage detection (LP is recognized directly from the image) or with two-stage detectors (first the vehicle is recognized, then the vehicle clipping is used for LPD). In both strategies, LPD can be viewed as a specialization of generic-purpose object detectors trained with LP images.

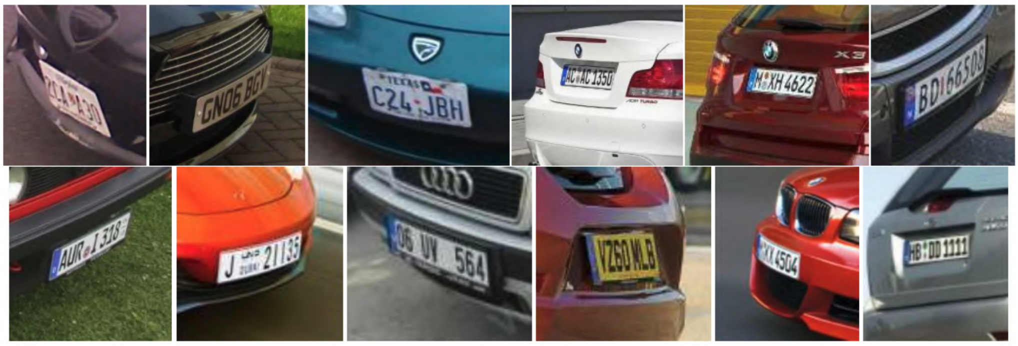
As an example of generic-purpose object detection that is commonly adapted to traffic applications, we can mention the YOLO family (REDMON et al., 2016), (REDMON; FARHADI, 2017), (REDMON; FARHADI, 2018) (WEN et al., 2022). The core idea is to treat detection as a regression problem and to associate class probabilities to each object in a single neural network. By adopting this strategy, YOLO networks provide fast predictions and are often used in real-time scenarios. Analyzing the YOLO versions over the years, we can notice several upgrades such as the two-stage detector (LI et al., 2017), new backbone proposals based on CSPNet (WANG et al., 2020a), new data augmentation methods (BOCHKOVSKIY; WANG; LIAO, 2020), and different improvements that make these networks widely used in the field of object detection and customized for ALPR systems. However, in this work, generic object detectors are not the focus, so we will not describe the aspects of these models deeply. We recommend Boukerche and Hou (2021) for more details.

In camera setups where oblique views of the LP are generated, the use of bounding box object detectors may result in a very coarse representation of LP. In these cases, OCR is also more challenging to perform when compared to mostly frontal views since the LP characters are distorted by perspective projection. The method presented in Silva and Jung (2018) addresses the entire ALPR pipeline and focuses on detecting distorted LPs. It consists of three main modules: (i) vehicle detection; (ii) license plate detection and unwarping; (iii) OCR. These three modules are explained in more detail next:

i) The YOLOv2 (REDMON; FARHADI, 2017) network was used for vehicle detection because it provides fast prediction and acceptable accuracy for the case of interest here. Since YOLOv2 is already able to classify a significant number of classes, including vehicles, no modifications were required to adapt the network for this module. Even though newer techniques for object recognition are available (PAL et al., 2021), we decided to use YOLOv2 for simplicity, since Silva and Jung (2018) provide a repository with free access to the code, and because of the robustness shown in the work mentioned. Besides that, the pipeline has the modules well defined and decoupled which helped the replacement and adaptation of new networks.

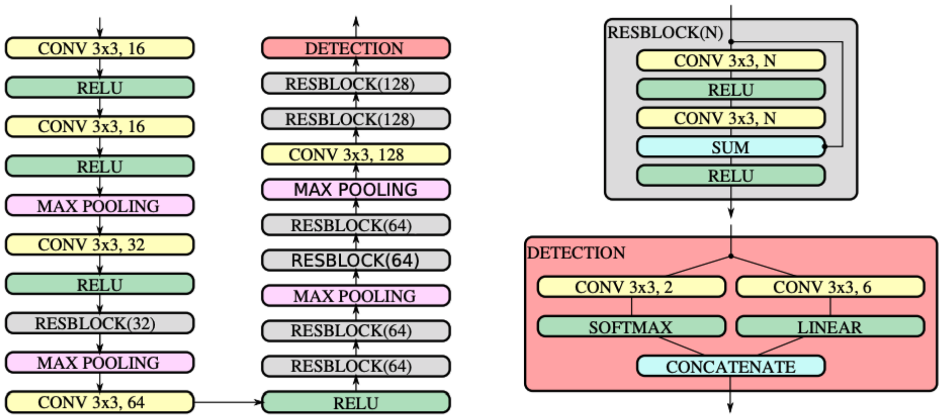
ii) When YOLOv2 indicates a positive result for a vehicle, the image is processed by WPOD-NET (Warped Planar Object Detection Network) (SILVA; JUNG, 2018), whose main goal is to identify the area of interest and represent it as an affine-transformed rectangle, represented as a quadrilateral. A planar homography transform can then be used to project the LP onto a frontal view. Figure 2.1 shows examples of LP oblique views, which are the main use cases of WPOD-NET.

Figure 2.1 – Oblique LP examples which are the main target problem that we aim to solve using WPOD-NET.



To accomplish this task, WPOD-NET uses a CNN (Convolution Neural Network) architecture with seven residual blocks and 21 convolutional layers. A residual block consists of connecting one layer to the next and also to the layers a few steps away. The convolutional layers, which give their name to this architecture, can be defined as a kind of hidden layer that applies a filtering function to the input data. The result of this operation is a feature. The deeper the convolutional layer, the higher the level of the extracted feature (HE et al., 2016a). Figure 2.2 shows the WPOD architecture in detail, giving an idea about all operations performed in a car image until extracting the LP rectangle.

Figure 2.2 – WPOD-NET architecture (SILVA; JUNG, 2018).



iii) After getting the LP in a frontal perspective, the final step is to identify the sequence of characters on the image. For this purpose, the authors used a modified YOLO network for OCR (MONTAZZOLLI; JUNG, 2017) trained using a dataset of synthetic LP images. The data was generated by applying random transformations to add noise, blur, and lighting variation to produce artifacts that might occur in real-world scenarios. To improve the performance in this type of task, a useful strategy can be applied that uses heuristics to determine the characters, since we have a well-defined pattern in license plates (LEE et al., 2010). In Brazil, for example, the current LP format has three letters, one number, one letter, and two numbers. Using this condition, we can then optimize the OCR results. Nevertheless, we will not focus on this strategy in this work.

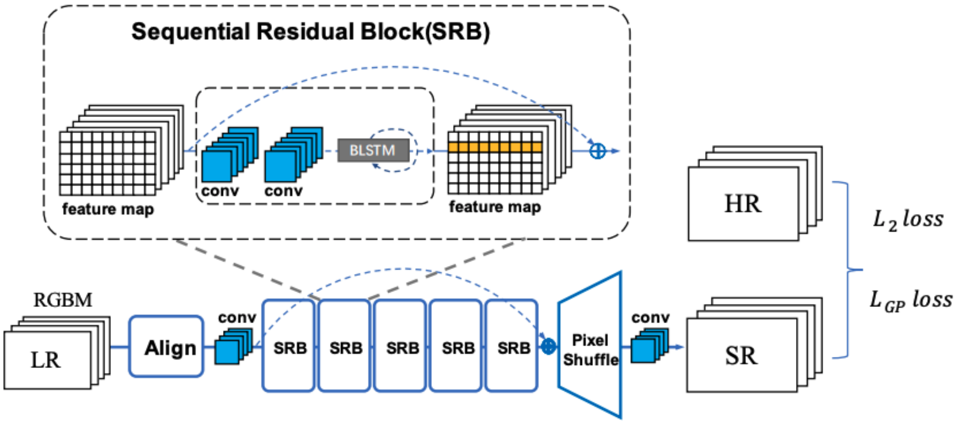
2.2 Image Super-Resolution

A super-resolution process aims to produce an image with better resolution – a high-resolution (HR) image – than its low-resolution (LR) version. There are several approaches to super-resolution: for example, image processing filters like bicubic interpolation were commonly used in the past (LIU; GAN; ZHU, 2013/11), (RUANGSANG; ARAMVITH, 2017). With the advance in deep learning techniques, CNNs achieved better results and led recent works in this area (ZHU et al., 2021; KIM et al., 2021). We can also see proposals of distinct loss functions to solve reconstruction problems like noise and lack of details in the generated image (LEDIG et al., 2017). To alleviate training times with very deep networks, residual blocks can be included to increase convergence rates, as we can see in Lim et al. (2017), Zhang et al. (2018).

There are also some approaches that focus on specific classes of images, such as those containing text. For example, a Text Super Resolution Network (TSRN) was proposed in Wang et al. (2020b), and it appears to be a good choice for improving the quality of low-resolution LP images. The main modules are:

i) use of sequential residual blocks, whose main idea is to take advantage of the sequential property of text images. Its implementation is based on the principle of recurrent neural networks (RNN) (HE et al., 2016b) and uses a residual block with a bidirectional LSTM mechanism. Figure 2.3 provides a high level description of this module.

Figure 2.3 – Text Super Resolution Network architecture (WANG et al., 2020b).



To understand these concepts, it is convenient to define the RNN architecture first. Unlike CNNs, where neurons in layer $N-1$ are connected to neurons in a layer N in a sequential manner, RNN connections can form cycles. Thus, the output of a particular neuron can influence the subsequent input of the same node, creating this temporal aspect that is useful in tasks such as speech or text recognition, or any other context that involves sequential input dependence. However, it has limits:

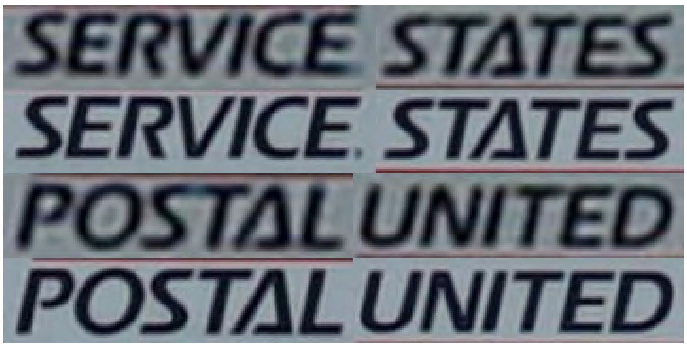
in speech recognition, for example, the gap between a pronoun and the subject is quite large, and the cyclic connections of RNNs are not sufficient to recover this dependency. For this reason, the Long Short-Term Memory (LSTM) architecture implements a powerful memory and is a suitable option to solve various problems where RNNs reach their limits. In addition to LSTM, there is bidirectional LSTM, which implements a similar mechanism to LSTM, but where information flows through cells in both directions, improving dependency recognition in sentences and speeches (HERNáNDEZ et al., 2019).

ii) Focus on the boundary reconstruction, a specific loss was designed to sharpen the character boundaries. Termed as "gradient profile loss", it has shown useful to better distinguish between the characters from the background and to obtain more evident shapes.

iii) Finally, a central alignment module is used to solve the problem of misalignment between pairs of LR and HR images. Implemented using a spatial transform network (STN) (JADERBERG et al., 2015) in combination with thin plate splines transformations (TPS), the alignment module can handle spatial variations in a flexible way and better align the corresponding pixels.

An important contribution of Wang et al. (2020b) was also the dataset used for training their model. Called TextZoom, the set of images contains pairs of real low- and high-resolution text scenes captured by cameras with different focal lengths and no specific environment specification like light intensity or orientation. Some examples can be seen in Figure 2.4. In this way, TextZoom simulates real cases better than LR datasets created from degraded HR images (also called synthetic low-resolution images). Although this most straightforward technique of creating low-resolution images from the HR version has been widely used to train and validate several Super-Resolution (SR) methods, the results obtained decrease significantly when applied to real scenes due to possible domain shifts. Nevertheless, datasets of synthetic low-resolution images were crucial for SR advance and are referenced in several papers demonstrating significant improvements in SR area using deep learning (DONG et al., 2016; KIM; LEE; LEE, 2016).

Figure 2.4 – TextZoom dataset samples: low and high resolution pairs of "SERVICE", "POSTAL", "STATES", "UNITED" words.

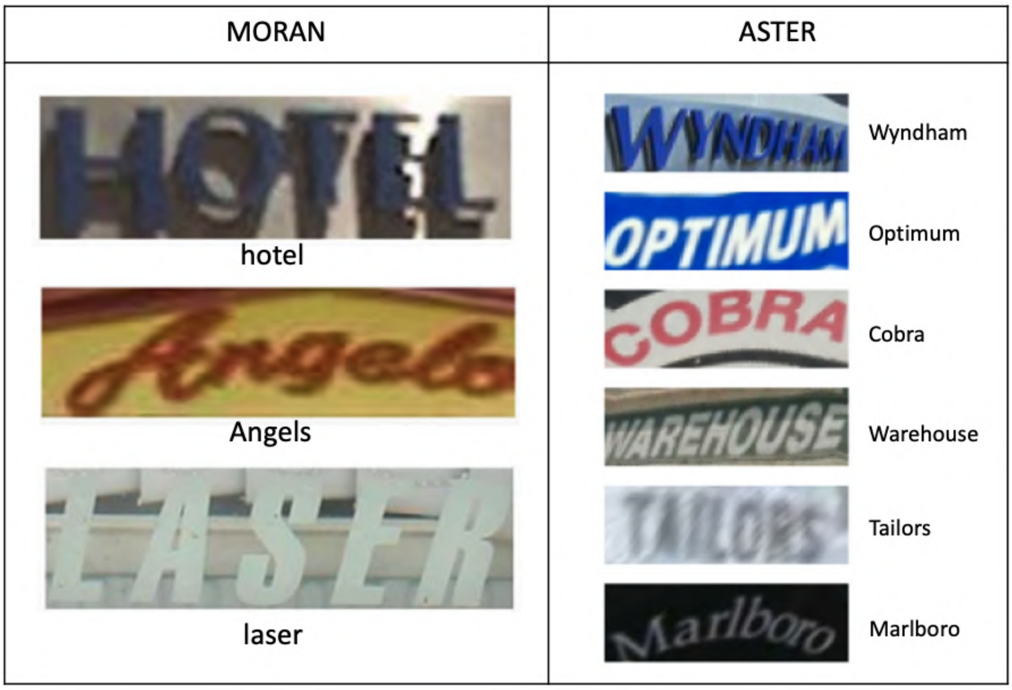


2.3 License Plate Recognition

Given the LP image, the final step of ALPR consists of obtaining the LP number, which is basically a string. Some methods, such as the ones chosen by Silva and Jung (2018), use object detectors to find individual characters and their respective categories (i.e., the letter or number) and then form the final string. However, this strategy requires bounding box annotations for each character which reduce the amount of data available and inject complexity into the labeling process.

On the other hand, there are several general-purpose OCR approaches that require only string-level annotations, such as MORAN (LUO; JIN; SUN, 2019) and ASTER (SHI et al., 2019), which are easier to annotate. In this type of mechanism, OCR usually works under the text scene using an attention-based method that simulates human cognitive attention: it focuses on a small (but most important) portion of the input data to extract information. Also in the case of the previously mentioned OCRs, Convolutional Recurrent Neural Networks (CRNNs) are used as base architecture and combined with ASRN (Attention-based sequence recognition network) and LSTM mechanisms to build the text recognition modules. Thus, MORAN and ASTER can predict an entire string focusing on each character in a different moment while considering the adjacent area with less attention. To deal with unaligned text, both networks include rectification modules and can, therefore, successfully recognize challenging text scenes as shown in Figure 2.5.

Figure 2.5 – Text scene samples used for ASTER and MORAN validation in (SHI et al., 2019; LUO; JIN; SUN, 2019).

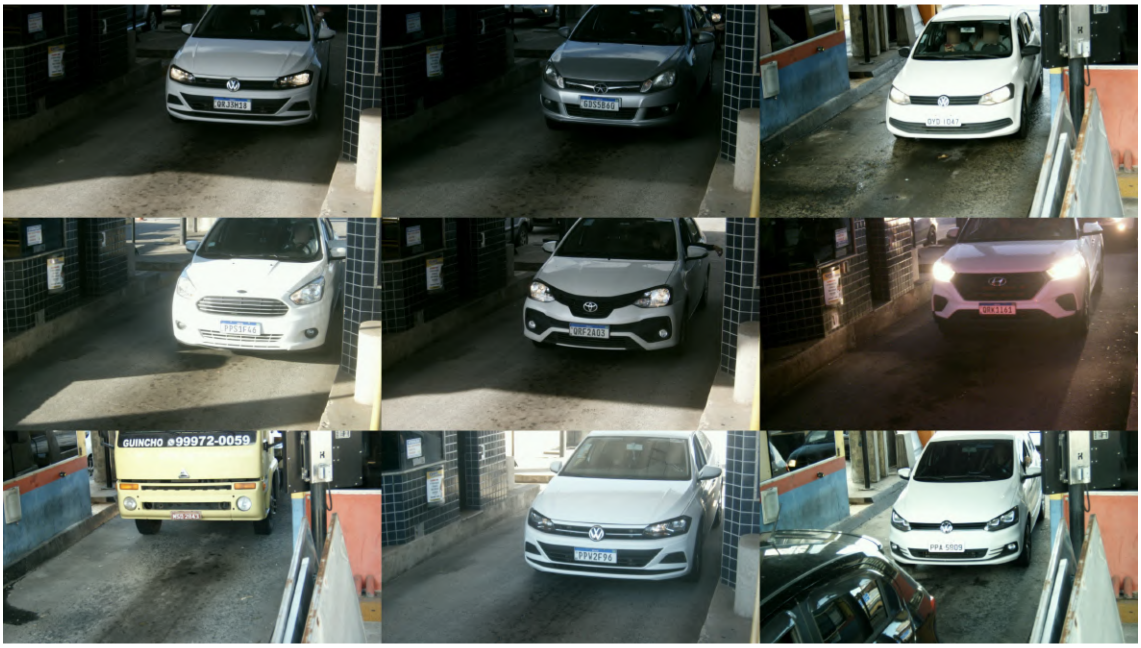


3 DATASETS

For this work, we used three datasets for training, validation, and cross-dataset testing, as we will see next in details. Since the main motivation for adding a text super-resolution module was to adapt the ALPR system to achieve better results in real scenes, the tests should be performed with real traffic scenes. In this sense, two of the datasets used for this work contain images taken with traffic cameras under different conditions. The other is the TextZoom dataset mentioned in Section 2.2, which was used to perform the initial TSRN training, since the use of pairs of real low and high-resolution images may be beneficial for the final ALPR system. The datasets are briefly described next.

- RodoSol (Laroca et al., 2022): this dataset contains 20,000 single-shot images of different cars, motorcycles, trucks and vans captured in pay tolls located in the Brazilian state of Espírito Santo. The dataset is organized in four categories: car images with license plates in legacy format, car images with license plate in Mercosul format [1], as well as motorcycles in both the legacy and the Mercosul format. Nonetheless, focusing on our case of study, a subset of 10,000 images has been created containing only car images balanced between both LP formats. In Figure 3.1, we can see image samples extracted from the RodoSol dataset.

Figure 3.1 – Image samples from RodoSol dataset.

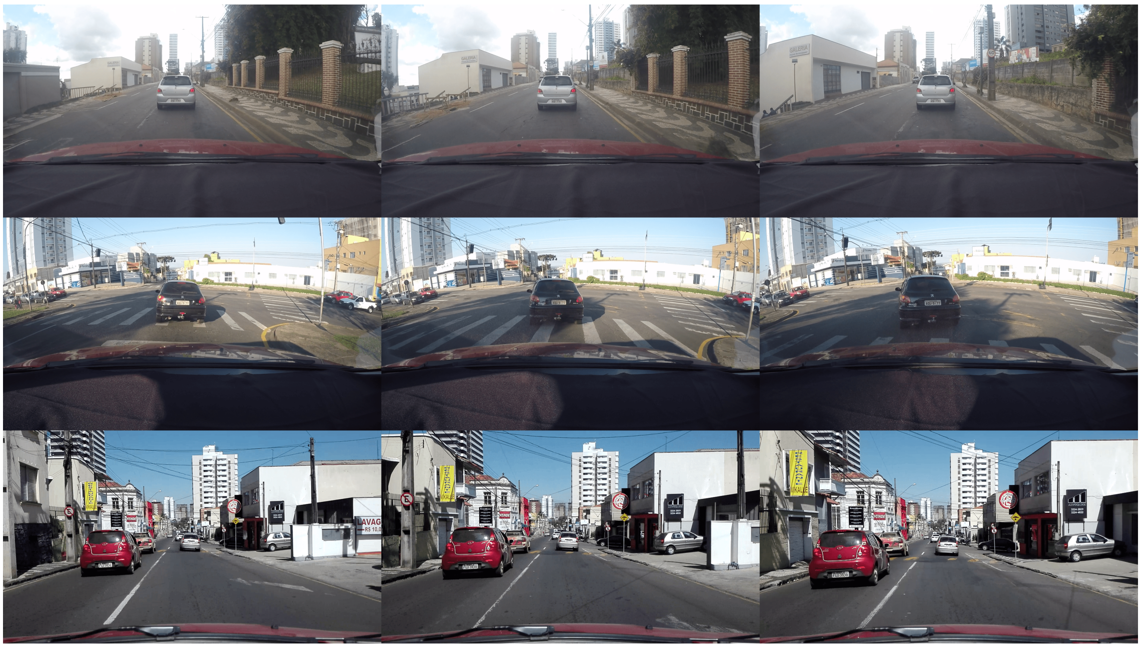


[1] The Mercosul license plate identification system was implemented in 2020 and it is a unified system that includes all the countries in the Mercosul economic block

- UFPR-ALPR (Laroca et al., 2018): this dataset was created using a car-mounted camera tracking vehicles on Brazilian roads. A total of 150 vehicles were tracked on different routes, taking 30 photos by car from many perspectives. The vehicles were cars, motorcycles, trucks, buses, and vans. However, for our study, we selected only car images creating a subset for testing with 2,815 samples. Figure 3.2 shows images of three vehicles taken in different positions on the road to illustrate the idea of the Rodosol dataset.

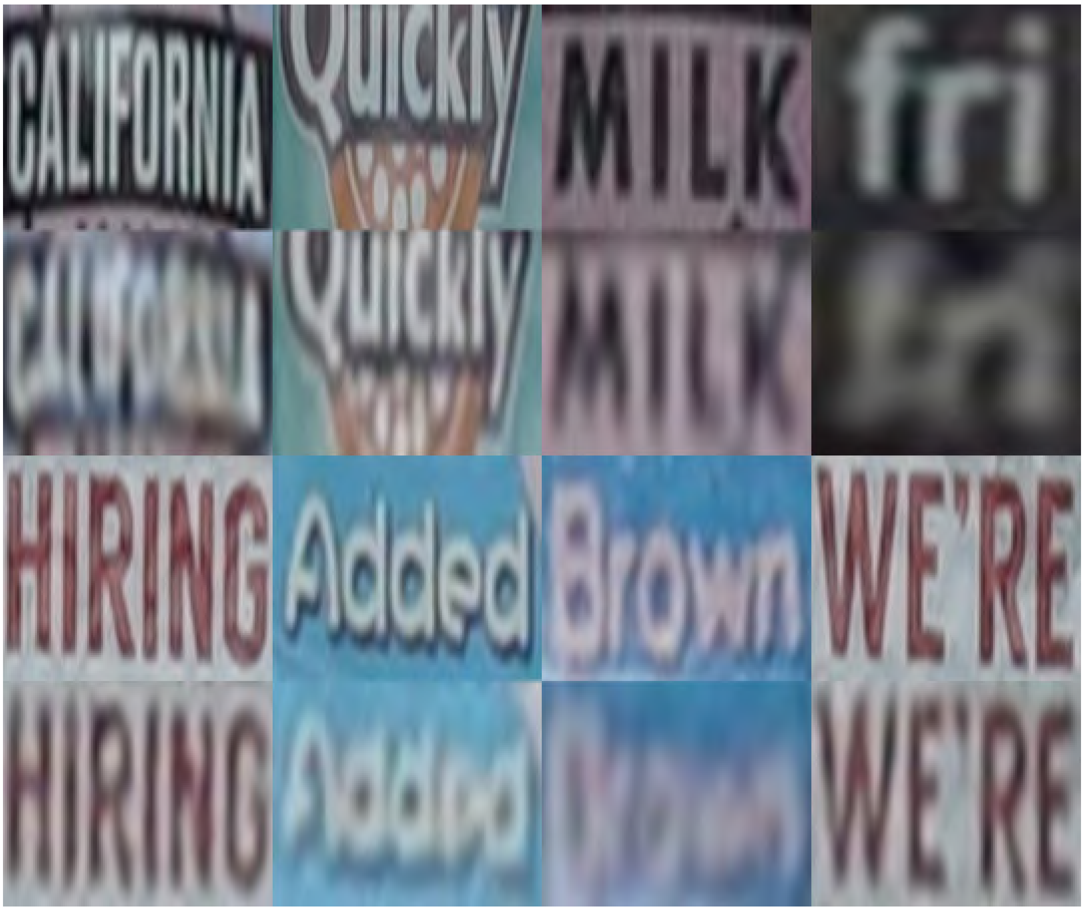
  Some features of this collection were considered to select it as a test dataset, such as the different distances between the car and the camera, resulting in cropped LPs at different resolutions. Another aspect was the multiple perspectives we have of the cars. As can be seen in Figure 3.2, we have different angles and views, which is perfect to verify that our ALPR system is independent of environment settings or camera setups. Both attributes are consistent with our proposed goals: dealing with low-resolution license plate images coming from generic capture setups.

Figure 3.2 – Image samples from UFPR-ALPR dataset.



- TextZoom (WANG et al., 2020b): unlike RodoSol and UFPR-ALPR datasets, the focus of TextZoom is not vehicles or license plates. This dataset was created for general text recognition purposes and contains text images in several font styles, sizes, rotation angles, and colors. In total, the dataset contains 21,740 text scene images organized in a subset for training with 17,367 images (around 80% of the collection) and a subset for testing with images classified in three different levels of difficulty: easy, medium and hard. Misalignment, font style and character ambiguity are the aspects used to classify an image in one of these three levels of difficulty. Besides that, these text scenes are true low- and high-resolution image versions, and that was our motivation for including this dataset in our training strategy.

Figure 3.3 – Image samples from TextZoom dataset.



4 THE PROPOSED METHOD

The combination of the two works described in the previous chapters –ALPR (SILVA; JUNG, 2018) and TSRN (WANG et al., 2020b) – is expected to create an improved ALPR system that can achieve higher accuracy when compared to the baseline ALPR system proposed by Silva and Jung (2018), particularly for low-resolution LPs. Since TSRN leads to better-defined boundaries around the character and other improvements mentioned in Section 2.3, the idea is to introduce this new component between the rectification and the OCR module, as illustrated by comparing the high-level architecture proposal by Wang et al. (2020b) in Figure 4.1 and the architecture containing the SR module in Figure 4.2. In other words, TSRN will be responsible for refining the LP image produced by the rectification module by alleviating low-resolution degradations and enriching the content to boost OCR performance.

Figure 4.1 – ALPR high level architecture proposed in Silva and Jung (2018).

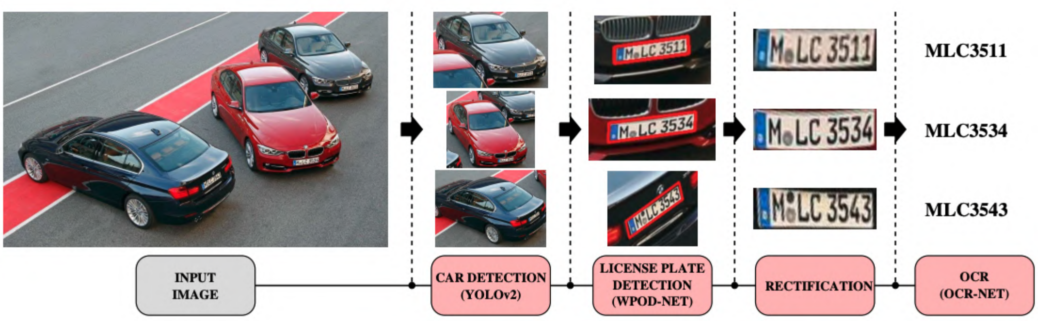
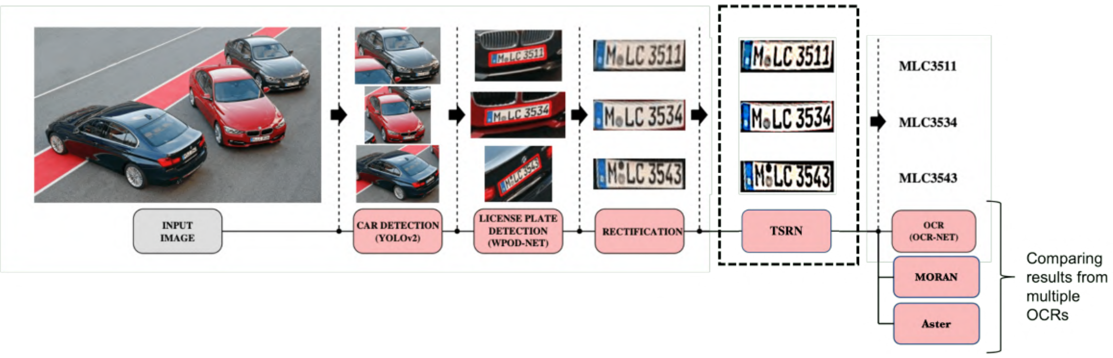


Figure 4.2 – High level architecture proposal representing the modified ALPR system including TSRN module.



Although using a pre-trained TSNR as a black box in the ALPR pipeline might be a natural choice, there are several related questions that we want to answer. For example, regarding the OCR modules, we want to test whether string-level OCR trained with generic text performs better than character-level OCR. One of the advantages of using string-level OCRs is the ability to explore the semantic patterns of words in a given language. For example, in English, the character “t” is more likely to be followed by an “h” than any other letter (JONES; MEWHORT, 2004), so the model can learn these patterns and use them as an advantage. Since there is no semantic dependency between letters in the LP context to explore, using character-level OCR seems reasonable. However, given recent advances in string-level models and the ease of dealing with string-level annotations, testing the performance of these models in the LP context could result in a feasible option. We also want to test whether fine-tuning a model for the LP context using a network previously trained with generic-text images works well in our ALPR system.

Regarding the TSRN, we also have some validation points about fine-tuning. The pre-trained SR network chosen for this work was previously trained with generic-text images. So, is fine-tuning required to achieve acceptable performance in the context of LP? Or is the TSRN able to adapt to the LP context using only generic-text images as the training dataset? The last question we want to answer with our experiments concerns the loss function used during the TSRN training phase. Since the purpose of TSRN is our application to produce an increased-resolution image that improves OCR inference, rather than an image that is visually better, it is logical that OCR performance affects TSRN loss. To test this assumption, one of our experiments includes training the TSRN module using a loss resulting from the combination of OCR and TSRN losses.

For evaluation, the dataset UFPR-ALPR was chosen because it contains a large number of cases that include images of cars at different positions and distances from the camera. In this way, we were able to analyze ALPR performance not only for low-resolution LP images, but also in cases where light variations or the presence of shadows obscuring the LP make character recognition difficult. As for the metrics used in this work, full OCR accuracy is the one that could guarantee the best usability of our system, and we consider one case as correct when the OCR output completely matches the LP alphanumeric sequence. That is, if one or more characters do not match the content of LP, it is considered an incorrect case penalizing the metric result. We chose this approach because, in a real-world use case, a single incorrect character is sufficient to invalidate the entire ALPR result, so the system is only useful when the entire sequence is correct.

On the other hand, the use of accuracy by character rather than by LP sequence might give a false idea of the actual performance of ALPR. This is because even if the ALPR output contains one incorrect character in all cases, the overall accuracy is about

85% (for a LP with seven characters). However, there is not a single case that we can consider correct from the perspective of real-world application, and this was the reason to choose accuracy by sequence to evaluate the ALPR proposed in this work, which gives a better sense of usability when applied in real scenarios.

4.1 Training and Evaluation Strategies

One of the goals of this work is the creation of an ALPR system that is independent of environment settings, i.e., a generic ALPR system capable of identifying LPs in several capture setups, including views that were not used during the training. To validate this capability, we performed a cross-dataset testing using the UFPR dataset only for testing, while RodoSol was used to train and validate both the OCR and TSRN. Since UFPR dataset contains different vehicle categories and our focus is on car images, we selected only the images of interest with a total of 2,815 captures, some of them related to the same vehicle but with a different view. This collection served as a test dataset used to check the impacts of our enhancements and it was not used to train the TSRN or any of the mentioned OCRs.

The experiments performed in this work are mainly related to two ALPR modules: the new super-resolution module and OCR. TSRN training and fine-tuning were necessary because our first tests after the introduction of the SR module did not reflect a significant enhancement of the ALPR system, which led us to think about the use of fine-tuning techniques to obtain better results. On the other hand, the widespread use of ASTER and Moran OCRs was a motivation to evaluate these OCRs in our new ALPR version.

Regarding the vehicle detection module, we note that most object detectors provide pre-trained weights in datasets that contain vehicles (such as VOC 2012 and COCO 2017), and any of them (such as Gao et al. (2019), Wen et al. (2022)) can be used. However, we decided to keep using the YOLOv2 implementation for simplicity. Therefore, as a suggestion for future works, we can replace the vehicle detection module with a more recent object detection network. Next, we present the training setup related to different modules of the ALPR pipeline.

4.2 TSRN

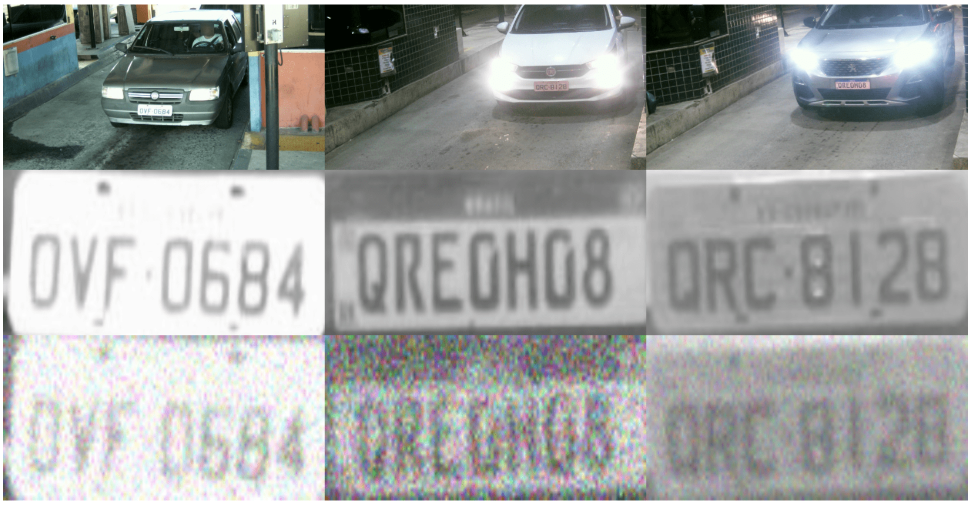
With respect to the datasets used to train TSRN, we continue to use TextZoom since the authors mentioned that using real low- and high-resolution image pairs instead of synthetic LR images was one of the main reasons to achieve better results in real scenes. Since such a dataset contains text images related to general-purpose applications, we explored transfer learning with LP images to direct our network to the area of interest without losing the advantage of the TextZoom dataset.

The TSRN training used the same hyperparameters provided in the original paper: learning rate of 0.001, downsampling scale of 2, 500 epochs and Mean Squared Error (MSE) as loss function. With this configuration and using pre-trained ASTER as OCR, we got an accuracy of 73.4% / 54.93% / 38.72% against the TextZoom (easy/medium/hard) collection.

Unfortunately, after some experiments described in more detail in Chapter 5, we found that fine-tuning is needed and that a collection of low- and high-resolution LP pairs is required for us to proceed. The lack of available datasets for this purpose led us to create our own set using the RodoSol dataset, the WPOD network, and simple downsampling algorithms. First, we extracted all LP images from the RodoSol dataset using WPOD net, assigning the rectangular area to each LP image. This allowed us to determine the best options for high-resolution LP images. The next step was to obtain the low-resolution pair by reducing the images by a factor of 2 and adding noise. The method used for the downscaling was interpolation, and we also added Gaussian noise and randomly changed the value of some pixels to simulate a degration like salt and pepper. The parameters such as the number of pixels to change and the noise standard deviation σ to calculate the Gaussian blur were also randomly generated, allowing us to create multiple low-resolution versions for each high-resolution image. In the end of this process, we had two datasets: the first one with the 1,000 and the second one with the 5,000 highest resolution images of LP extracted from RodoSol.

In Figure 4.3 we can see some examples of images used as high-resolution and one of the possible low-resolution versions. Since this synthetic dataset was used only to fine-tune our net, we hope that the weights do not suffer extreme changes and lose the benefits of TextZoom.

Figure 4.3 – RodoSol image examples used as high and low resolution pairs to fine-tune TSRN.



4.3 OCR

In Silva and Jung (2018), the OCR task was performed by training an object detector for each character, as mentioned in Section 2.3. However, since MORAN and ASTER achieved remarkable results on general text scene images even in cases of oblique perspectives, it would be an interesting test to evaluate these networks in LP recognition scenarios, and this was the motivation for some of experiments described in Chapter 5. In addition, string-level annotations are easier to obtain than annotating each individual character allowing us to build a dataset of LPs to fine-tune string-level OCR models for this context.

Since ASTER achieved the highest performance among the OCR models tested, we selected this model to specialize using transfer learning techniques to recognize LP and replace the OCR module in the final ALPR version. The details of this and the previously mentioned experiments are presented in Chapter 5.

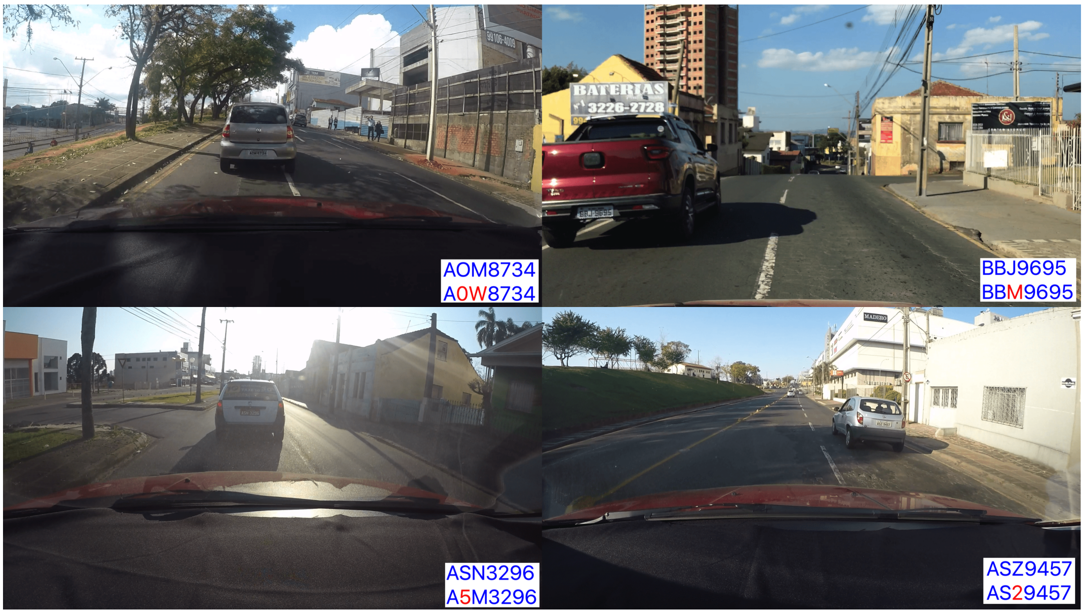
5 EXPERIMENTS AND RESULTS

In order to analyze each of our proposals, we have defined a series of experiments to modify and test the new modules in an incremental and controlled way. For each experiment, we also compute the accuracy over the test dataset as a method to evaluate whether the change resulted in a better ALPR version.

5.1 Experiment 0

The first experiment was to define our starting point, which was the original ALPR presented in Silva and Jung (2018) without the SR module and using the character-level OCR. With the baseline approach, we achieved 53.57% of accuracy (a result is considered correct when all characters in the LP are correctly recognized), which presents potential to improve. Some examples where the actual ALPR did not perform well are shown in Figure 5.1, where the ground truth and the OCR output are shown in the bottom right of each image.

Figure 5.1 – Result samples from experiment 0 with expected and predicted OCR result.



As we can see, the original ALPR achieves reasonable accuracy, but there are still a considerable number of cases that we need to work on and try to improve. In the next experiments, we will apply the previously proposed changes aiming to achieve higher accuracy over the same testing data.

5.2 Experiment 1 and 2

The next two experiments were related to the OCR module. In this step, we modified the original OCR, which is based on object detection using the YOLO architecture, and replaced it with ASTER (SHI et al., 2019) and MORAN (LUO; JIN; SUN, 2019) networks. Both works provide a pre-trained network that can be easily used – just to have an idea about the previous results of these networks, ASTER achieved an accuracy of 89.5% in the SVT dataset[1], while Moran was able to get an accuracy of 88.3% in the same data collection. In other words, both networks excel when we are dealing with generic text recognition. In that way, since we are using the pre-trained networks for generic text recognition, we will analyze if these models are able to have a similar performance when applied in LP context, in which text images have different aspects than the generic text scenes used to train these networks.

However, against the UFPR dataset, the ALPR using pre-trained ASTER and MORAN as the OCR module achieved 23.26% and 1.42% accuracy, respectively, indicating that the pre-trained model adapted poorly to the LP domain. As mentioned before, these models were trained for generic text recognition where we have several differences when compared to LP context, especially regarding the presence of digits between alphabetic characters, which is a very rare case in generic text datasets. Some examples of ASTER and MORAN outputs are shown in Figure 5.2.

Although both pre-trained models did not achieve good results, we selected ASTER to fine-tune and test whether it could be considered as a replacement for the original OCR module. The dataset used in this phase was created using the car images from the RodoSol collection and the WPOD network for LP extraction. The result was a set of cropped LP images that can be used as input for ASTER fine-tuning.

After this process, we got 18,464 LPs images (some examples are shown in Figure 5.3), which were split using a rate of 80% for training and 20% for validation. Table 5.1 shows the accuracy progress in the validation set over epochs starting from 321 (the last pre-trained epoch) to epoch 351, when the accuracy reaches the saturation level over validation dataset. The same data is shown as a plot in Figure 5.4, which makes clear the saturation pattern.

[1] The Street View Text Dataset (SVT) is a collection of images extracted from Google Street View containing outdoors and storefronts with general text. Repository: <http://vision.ucsd.edu/~kai/svt/>

Figure 5.2 – Result samples from experiments 1 and 2 using ASTER and MORAN in the OCR module.

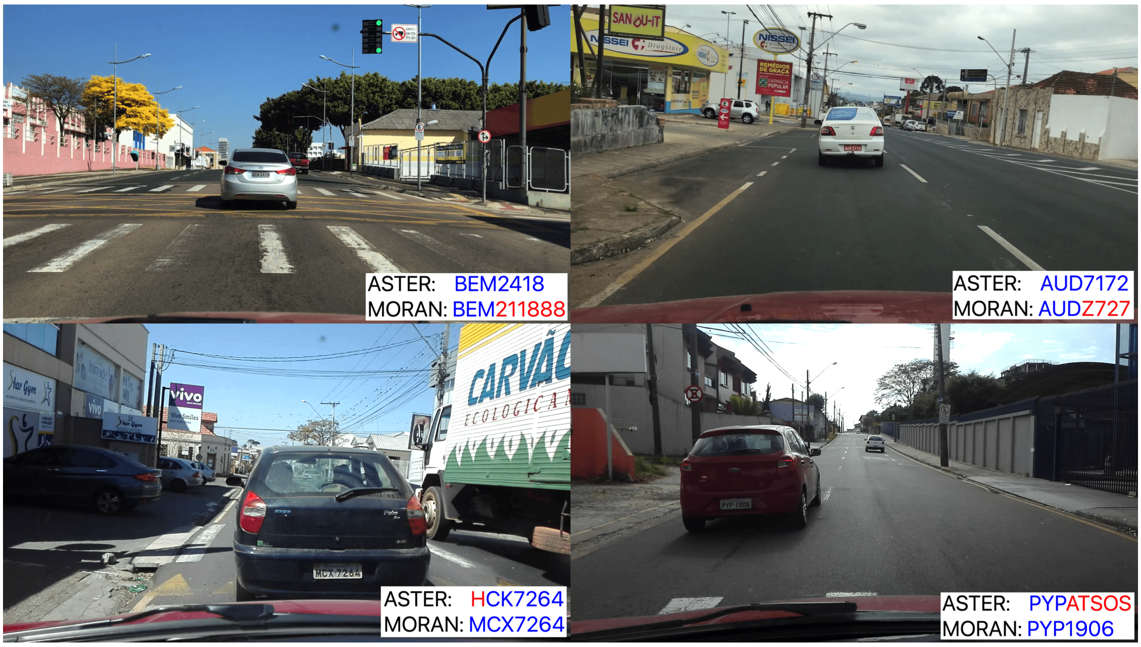
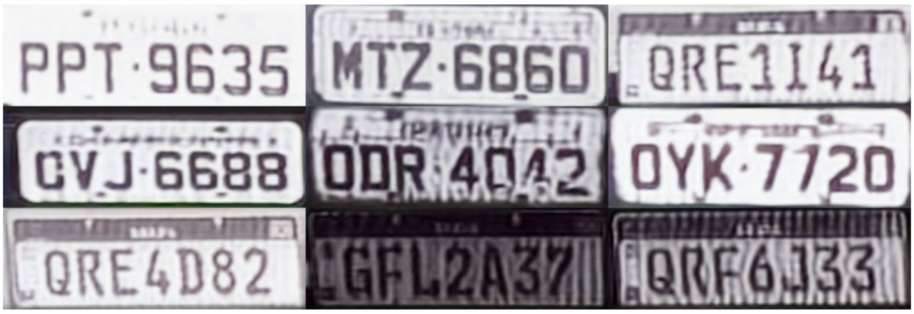


Figure 5.3 – LP images obtained from RodoSol dataset.



5.3 Experiment 3

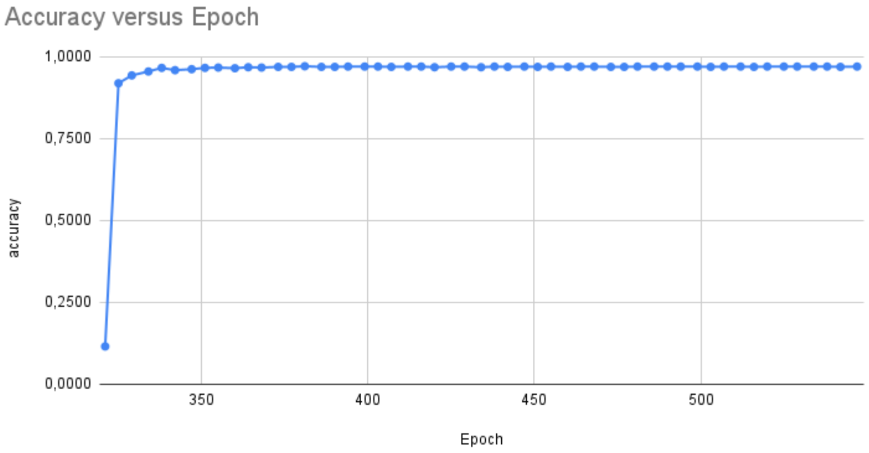
Once using the ASTER pre-trained with generic text scenes did not result in acceptable performance, in this experiment, we tried the fine-tuned ASTER model in the OCR module. In that way, we will be able to answer if a string-level character is a suitable option to recognize LPs or if the lack of semantic aspects of this context makes hard OCRs of this type adapt.

Running the experiment with ASTER fine-tuned, we can see that ALPR achieves 88.34% accuracy. This is 34.77% higher than the value obtained using the original OCR and 65.08% compared to the pre-trained ASTER, which means that the fine-tuning process worked very well. Also, it makes clear how important OCR performance is in this

Table 5.1 – ASTER training - Epochs versus Accuracy.

Epoch	*Accuracy*
321	0.1140
325	0.9190
329	0.9430
334	0.9550
338	0.9660
342	0.9590
347	0.9620
351	0.9660

Figure 5.4 – ASTER training - Epochs versus Accuracy



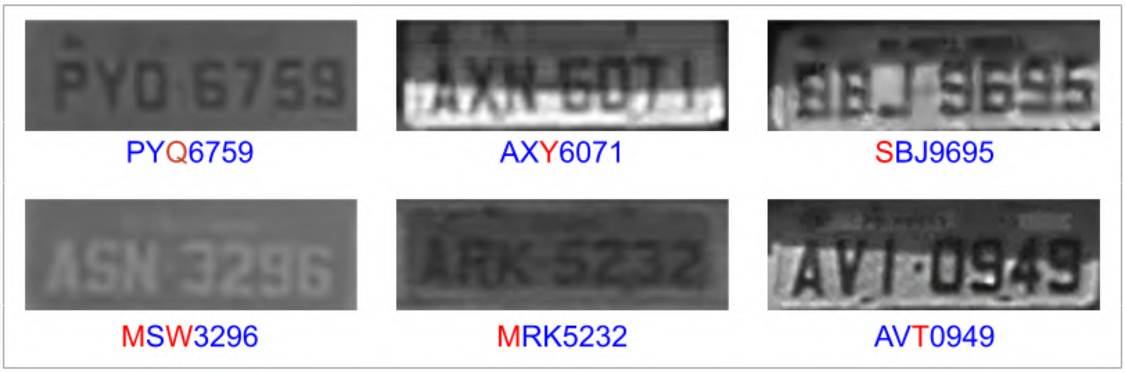
system because even with a high-quality image, a less than optimal OCR drastically affects the overall result.

In Figure 5.5, we can see some error cases produced by ASTER, where the character marked in red is the incorrect value produced by the network.

5.4 Experiment 4

In this experiment, the super resolution module was introduced in the ALPR pipeline for the first time, allowing us to get an impression about its performance on LP context. The TSRN model used was trained only with the TextZoom dataset, so no specialization for our interested context was done at this moment. In that way, the goal of this experiment is also to analyze whether a fine-tuning is also required to TSRN as such it was for ASTER.

Figure 5.5 – Result samples from experiment 3.



As mentioned, we will use TSRN when the LP image is in low-resolution, so the metric used to identify when we are dealing with this particular case is the LP area. Our OCR was trained expecting an LP size of 240×80, which is set as the target output resolution in WPOD in the final rectification process regardless of the original resolution of the unwarped LP. In this experiment, we first identify if the unwarped LP presents a low resolution, and in this case we set the target resolution for the unwarping process in WPOD to 120×40. This is done to avoid introducing artifacts in the interpolation done by WPOD, and then use TSRN to reach the desired resolution for OCR. In the other cases where TSRN is not required (i.e., the detected LP is already large enough), the WPOD output will be already in the size of 240×80.

To classify an LP as a low resolution case, we selected three different values to test: 120×40 (area of 4,800 pixels), 96×32 (area of 3,072 pixels) and 72×24 (area of 1,728 pixels). To reach these values, we reduced each dimension of our first experimented value (120×40) by 20% and 40%, so no scale disruption was introduced to the images. In that way, in our first test, all LP imagens with an area lower than or equal to 4,800 pixels will be resized to an image of 120×40 and classified as a low resolution case that needs to be processed by SR module. The same approach was used to test the other area thresholds.

For this first experiment with TSRN, no fine-tuning was used and the parameters for the training were presented in Section 4.2. In Table 5.2, we can see the accuracy and the number of low resolution cases for each LP area value. As we can see, the ALPR accuracy did not increase significantly when applying TSRN and OCR . In the best case, we had an increase of 0.22%, which represents a very small improvement. Figure 5.6 shows some cases where the SR module helped the OCR, and, in the right column, the output value provided by the version without the TSRN. On the other hand, Figure 5.7 shows cases where the SR module worsened the OCR performance, which means cases where the OCR missed the LP string after SR was introduced (right column shows the OCR output after TSRN introduction).

Table 5.2 – Experiment results for different low resolution thresholds.

LP area threshold	*# LR cases*	*LR cases %*	*ASTER accuracy*
120 × 40	2051	72.85%	88.56%
96 × 32	1419	50.40%	88.24%
72 × 24	589	20.92%	88.41%

Figure 5.6 – Cases in which the introduction of SR affected positively.

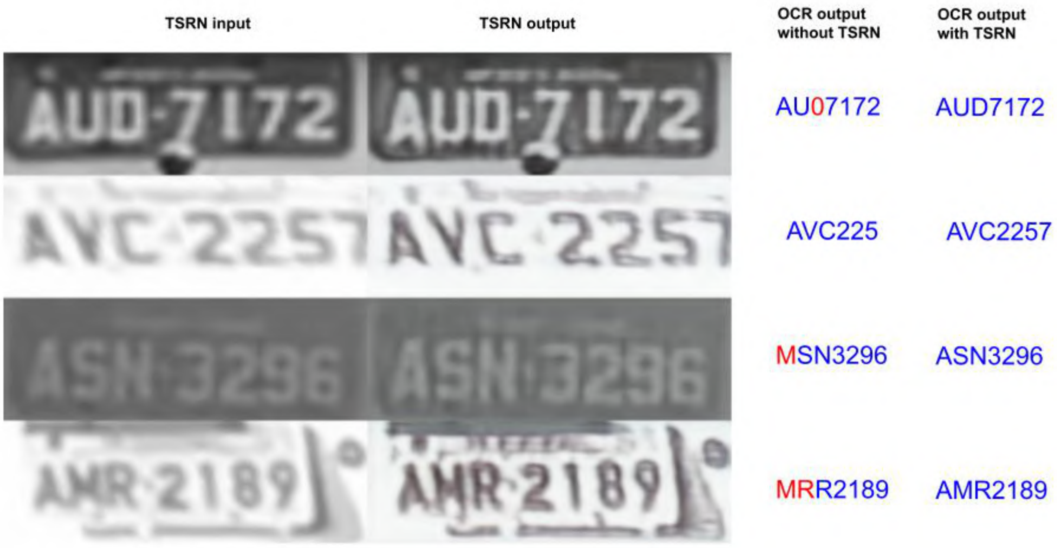
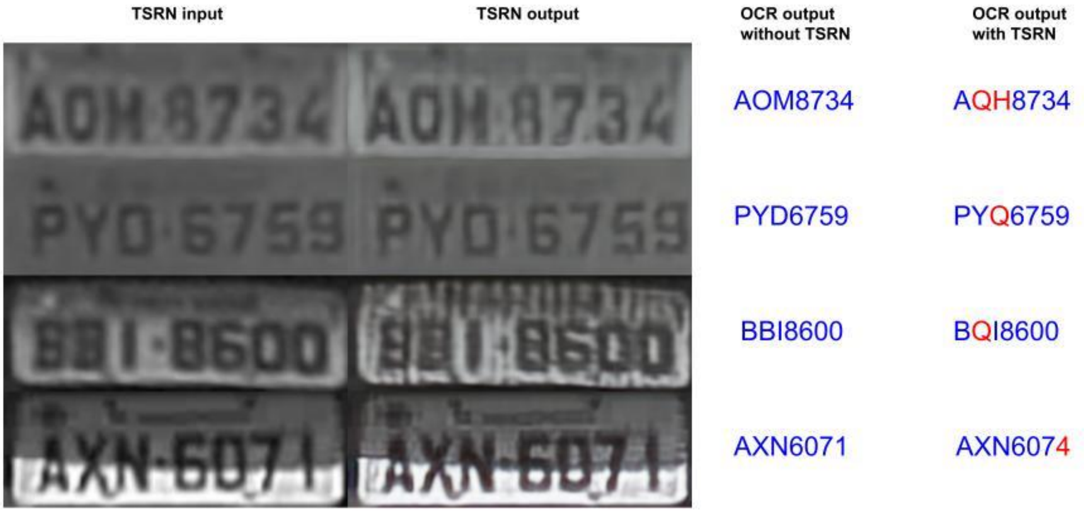


Figure 5.7 – Cases in which the introduction of SR affected negatively.



Since testing different thresholds for classifying an image as a case to be processed by the SR module showed no significant effect on accuracy, we will use the value of 120×40 for the next experiments.

5.5 Experiment 5

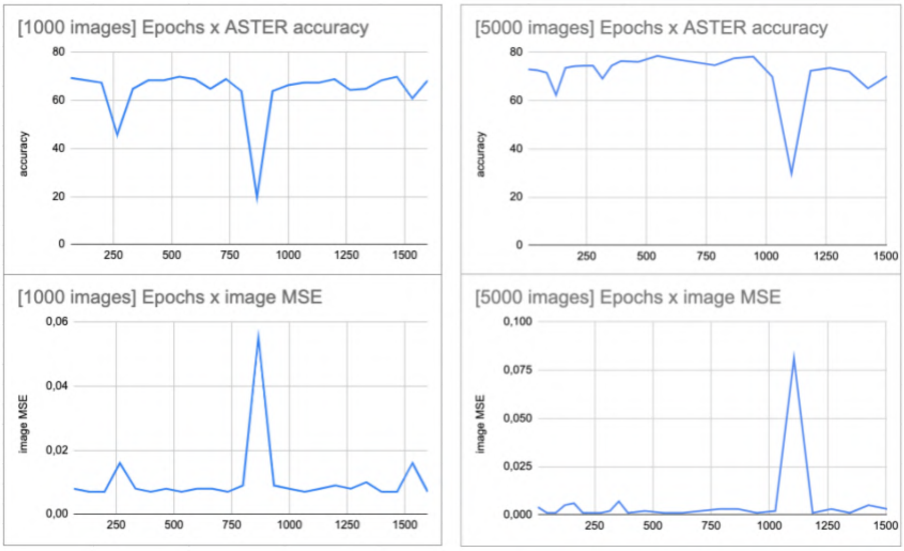
Once the fine-tuning worked satisfactorily for ASTER network, we performed the same for TSRN aiming the SR module to result in a more relevant gain over the overall accuracy. Also, it will allow us to answer if the lack of LP images in the training dataset used in 5.4 was the reason for the low gain after the introduction of the SR module. For the fine-tuning process, the synthetic LP dataset described in Section 4.2 was used running two experiments with the subsets of 1,000 and 5,000 LR and HR image pairs. Each of the subsets were split in training and validation parts using a rate of 80%/20%. The loss function keeps the same presented in the TSRN paper: the MSE over the target and output images, defined by

$$L_1 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{5.1}$$

where Y_i is the high-resolution version of an image in the training set and $\hat{Y}_i$ is the super-resolved image produced by the network.

As for fine-tuning, we did not freeze the weights of any layer allowing all values to be adjusted to fit this new dataset, and we monitored the training by measuring the ASTER accuracy across epochs. The result can be seen in Figure 5.8, where the best ASTER accuracy in validation image collection was 69.85% and 78.54% for training using 1,000 and 5,000 images, respectively, while the MSE average for each group was 0.01045 and 0.0080. Visually, the TSRN did a good job removing the synthetically added noise, as we can see in Figure 5.9, but it does not seem to be sufficient to boost the accuracy of the ASTER module, which did not change as expected. Testing this version of ALPR on our test dataset (UFPR-ALPR), we obtained an accuracy of 88.04% with the TSRN fine-tuned with 1,000 images and 88.48% with the version fine-tuned with 5,000 images which is not a relevant change compared to the non-fine-tuned TSRN from Experiment 4.

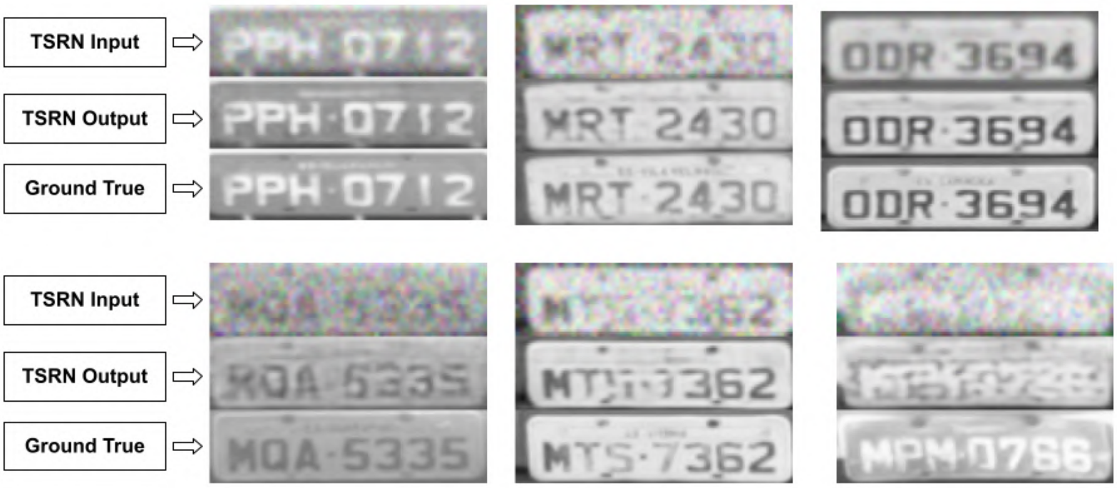
Figure 5.8 – ASTER fine-tuned: Epochs versus ASTER accuracy versus image MSE.



5.6 Experiment 6

An explanation for the stagnant accuracy even after TSRN fine-tuning could be related to subtle or even non-human visible image attributes that are actually explored by the OCR module. An image consists of a set of features such as shapes, colors, boundaries, curves, and patterns from which humans can infer some information. As we can see in Figure 5.9, TSRN does a good job recovering these features in a way that is visually comfortable to us. However, the features that are important for OCR might be quite different from those needed for human perception, and this might be the reason why OCR does not perform better even at a low MSE.

Figure 5.9 – TSRN processing results.



With this in mind, this experiment will test a new loss function that takes into account both the MSE between the target image and the TSRN output and also the ASTER loss function called Sequence Cross Entropy proposed in (SHI et al., 2019), which directly relates to the OCR task. If the component y_t is the ground truth text represented by a character sequence and ρ_{ltr}, ρ_{rtl} are the predicted distributions of the left-to-right and right-to-left decoders, respectively, the loss is given by

$$L_2 = -\frac{1}{2}\sum_{t=1}^{T}(\log \rho_{ltr}(y_t|I) + (\log \rho_{rtl}(y_t|I)). \tag{5.2}$$

By combining both loss components (MSE and Sequence Cross Entropy), we aim to simultaneously keep the visual quality of super-resolved images (MSE term) and increase the accuracy of OCR (cross entropy). For this, normalized values were used and the combination was performed via weighted averages through

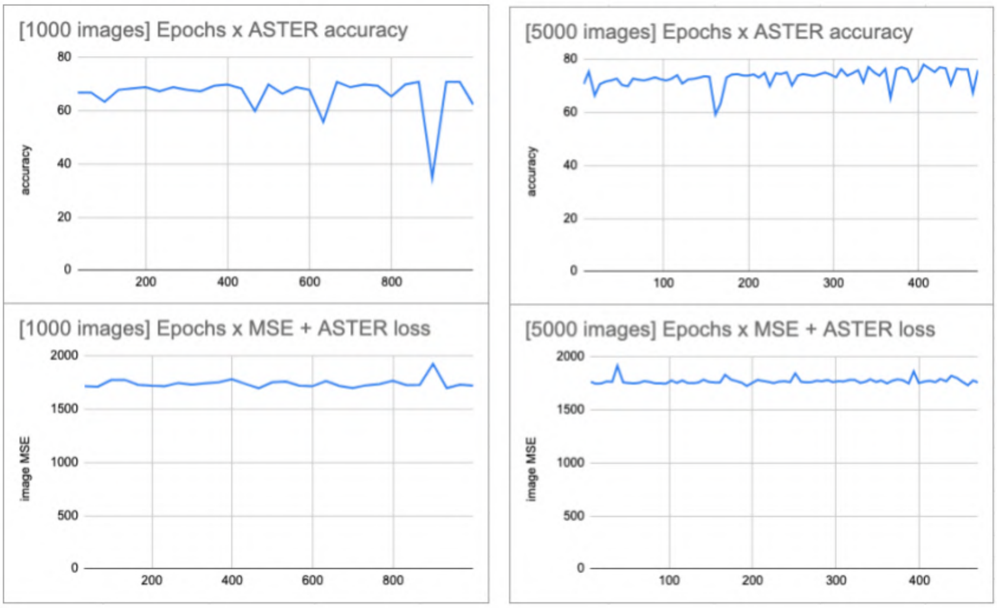
$$L = tL_1 + (1-t)L_2, \tag{5.3}$$

where $t \in [0,1]$ denotes the mixing weight. Using this combined loss, we re-trained the TSRN model using subsets of 1,000 and 5,000 images and obtained the results shown in Figure 5.10.

The last step of this experiment consisted of testing the TSRN models in the test dataset (UFPR dataset), but the obtained results indicated a marginal gain. The best accuracy was obtained using the weighting of 20% for MSE and 80% for Sequence Cross

Entropy (ASTER loss) ($t = 0.2$), and it was 88.63%, only 0.29% more than the ALPR without the SR module, which represents a very small increase.

Figure 5.10 – ASTER fine tuned using MSE and Sequence Cross Entropy loss function.



5.7 Summary of the results

Table 5.3 summarizes the experiments with the accuracy achieved at the end of each process for the test data set. It is worth highlighting the considerable gain obtained in Experiment 3 with the fine-tuned model ASTER, which exceeds our expectations, and the result of Experiment 6, which is the final ALPR version of this work.

Table 5.3 – Experiments summary

Experiment	*TSRN*	*OCR module*	*Accuracy over test dataset (UFPR)*
0	No super resolution network	YOLO OCR presented in (SILVA; JUNG, 2018).	53.57%
1	No super resolution network	ASTER available pre-treined network.	23.26%
2	No super resolution network	MORAN available pre-treined network.	1.42%
3	No super resolution network	ASTER fine tuned network using synthetic low resolution LP images built from RodoSol dataset.	88.34%
4	TSRN trained using the default parameters over 500 epochs and using TextZoom dataset	ASTER fine-tuned network using LPs extracted from RodoSol dataset.	88.56%
5	TSRN trained using the default parameters over 500 epochs and using TextZoom dataset + fine tune using synthetic low resolution images from RodoSol dataset	ASTER fine-tuned network using LPs extracted from RodoSol dataset.	88.48%
6	TSRN trained using the default parameters over 500 epochs and using TextZoom dataset + fine tune using synthetic low resolution images from RodoSol dataset and a loss function which consider the ASTER output	ASTER fine-tuned network using LPs extracted from RodoSol dataset.	88.63%

6 CONCLUSIONS

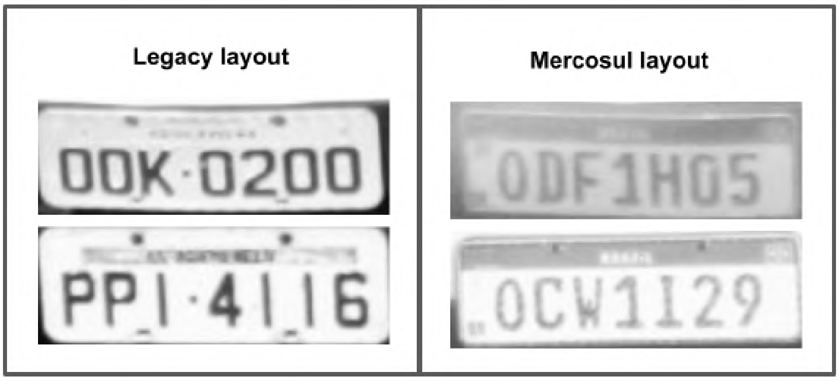
In this work, we tried two different approaches to increase the ALPR accuracy: adding a super-resolution module (TSRN) and changing the OCR to a generic and widely used one. During the experiments, we could identify needs and try new strategies aiming to get higher accuracies. Besides that, aligning with our proposed goals, we implemented a cross-dataset testing using two totally different image collections to ensure our system's flexibility. Regarding TSRN, we also tried a new loss function that considers not only the visual quality of the produced images, but also the accuracy of an OCR proxy task. In total, our changes on the ALPR increased 35.06% over the test dataset while OCR modifications were the most significant for this result as we can see in Table 5.3.

Even though TSRN performs well in the sense of building images with reduced noise and without resolution disruption, for the tested cases, the focus on the OCR module was what produced better advances and, probably, the module that needed more adjustments to increase the accuracy over the tested cases. This conclusion was motivated by analyzing the recognition errors, which frequently involve similar visual characters like O/Q, S/2, N/W, I/T, B/S, so training the OCR to improve the recognition of these cases could be a way to increase the final accuracy. A suggestion of strategy to handle these challenging cases could be adjusting the OCR loss to consider a higher value of gain or penalization when the case involves similar characters like the mentioned ones. In this way, the model must pay more attention to these cases over the training and be able to identify the small details that can be used to differentiate these characters. Besides that, the possibility of visually identifying these details that make the correct identification possible is an improvement which is more related to the OCR than the SR module, so, until this point, it is possible to assume that the goal of TSRN was achieved.

However, even if we bring OCR to the state of the art, we still have ambiguities that are inherent to Brazilian LPs. As shown in Figure 6.1, the letter "O" and the digit "0" are visually identical in the legacy layout. The same situation happens with the letter "I" and the digit "1". We know the difference because there is a well-defined pattern in Brazilian LPs[1] in the Brazilian LP system based on the position of the characters. So, in addition to improving OCR performance, a post-processing mechanism that implements the logic of the LP pattern is needed to achieve higher accuracy. Note that the visual ambiguity problem exists exclusively in the legacy layout and has been solved for Mercosul LPs.

[1] Assuming "D" denotes digit and "L" a letter, the Mercosul layout is defined as "LLLDLDD" while legacy layout is "LLL-DDDD".

Figure 6.1 – Ambiguity problem on legacy LP layout.



Another interesting result was the performance obtained with the fine-tuned version of ASTER compared to the pre-trained version. According to the authors, the pre-trained version represents the state-of-the-art when it comes to generic text recognition, and the results obtained in the SVT dataset confirm that. However, since the nature of the LP images is completely different, we expected a drop in performance, but not an accuracy of only 23.26%. Certainly, the SVT images are more complex due to the different spatial transformations, font styles and colors, even though the same model achieved 89.5%. The same behavior was observed with MORAN, which achieved lower accuracy in the well-behaved LP images, but remarkable results in SVT collection.

One explanation for this regards the semantic aspect in both datasets. While the SVT images relate mainly to English words, the LP cases have no semantic pattern and present combinations of letters and numbers in a way never seen in the SVT dataset. This also explains why the fine-tuning worked so well, because, after adding the never seen cases, the model recovered its performance.

About TSRN, one important comment regards the special LP cases like LPs with red background and cases with a high lighting variation or shadows (some of these cases are represented in Figure 5.7). Due to the small representation of these samples in our dataset when compared to the total image number, it leads us to an unbalancing problem and their presence becomes frequent between the missed ones.

Besides that, a not expected behavior tackles the TSRN loss function tested in Experiment 6, which combines the MSE between the images and the sequence cross entropy returned by the ASTER model. This approach was expected to increase accuracy because the TSRN would be shaped in a way to optimize the prediction of ASTER. However, as noticed in the experiment, there were no significant changes, which indicates that the low resolution is not the main problem that needs to be addressed to increase the achieved accuracy of 88.63%.

The analysis of the final results suggests that the replacement of the OCR module by the ASTER generic text recognition network exceeded expectations, as the accuracy in the validation dataset increased greatly and the TSRN played a supporting role. Nevertheless, this does not mean that the network was able to perform its job well in the pipeline. As can be seen in Figure 5.9, TSRN achieved good results recovering not only low-resolution degradations, but also removing noise and missing data. Another important point that we were able to confirm concerns the robustness of ALPR. After successful cross-dataset testing, we can say that the proposed pipeline is flexible enough to adapt to different environments and is able to detect, extract and recognize LPs even in images generated with different camera settings and perspectives. Moreover, the fine-tuning techniques applied during the tests proved their efficiency when it came to adapting OCR and TSRN to our area of interest. As we could see in Figure 5.4, only a few epochs were needed to saturate the loss.

REFERENCES

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. **YOLOv4: Optimal Speed and Accuracy of Object Detection**. arXiv, 2020. Available from Internet: <https://arxiv.org/abs/2004.10934>.

BOUKERCHE, A.; HOU, Z. Object detection using deep learning methods in traffic scenarios. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 54, n. 2, mar 2021. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/3434398>.

DONG, C. et al. Image super-resolution using deep convolutional networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 2, p. 295–307, 2016.

GAO, M. et al. Rgb-d-based object recognition using multimodal convolutional neural networks: A survey. **IEEE access**, IEEE, v. 7, p. 43110–43136, 2019.

HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

HE, P. et al. Reading scene text in deep convolutional sequences. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 30, n. 1, 2016. Available from Internet: <https://ojs.aaai.org/index.php/AAAI/article/view/10465>.

HERNáNDEZ, F. et al. Human activity recognition on smartphones using a bidirectional lstm network. In: **2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)**. [S.l.: s.n.], 2019. p. 1–5.

JADERBERG, M. et al. Spatial transformer networks. In: CORTES, C. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2015. v. 28. Available from Internet: <https://proceedings.neurips.cc/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf>.

JONES, M.; MEWHORT, D. Case-sensitive letter and bigram frequency counts from large-scale english corpora. **Behavior research methods, instruments, computers : a journal of the Psychonomic Society, Inc**, v. 36, p. 388–96, 09 2004.

KIM, J.; LEE, J. K.; LEE, K. M. Accurate image super-resolution using very deep convolutional networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

KIM, S. et al. Single image super-resolution method using cnn-based lightweight neural networks. **Applied Sciences**, v. 11, n. 3, 2021. ISSN 2076-3417. Available from Internet: <https://www.mdpi.com/2076-3417/11/3/1092>.

Laroca, R. et al. On the cross-dataset generalization in license plate recognition. In: **International Conference on Computer Vision Theory and Applications (VISAPP)**. [S.l.: s.n.], 2022. p. 166–178. ISBN 978-989-758-555-5. ISSN 2184-4321.

Laroca, R. et al. A robust real-time automatic license plate recognition based on the YOLO detector. In: **International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2018. p. 1–10. ISSN 2161-4407.

LEDIG, C. et al. Photo-realistic single image super-resolution using a generative adversarial network. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017.

LEE, Y. et al. License plate detection using local structure patterns. p. 574–579, 2010.

LI, Z. et al. **Light-Head R-CNN: In Defense of Two-Stage Object Detector**. arXiv, 2017. Available from Internet: <https://arxiv.org/abs/1711.07264>.

LIM, B. et al. Enhanced deep residual networks for single image super-resolution. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2017.

LIU, J.; GAN, Z.; ZHU, X. Directional bicubic interpolation — a new method of image super-resolution. In: **Proceedings of 3rd International Conference on Multimedia Technology(ICMT-13)**. Atlantis Press, 2013/11. p. 463–470. ISBN 978-90-78677-89-5. ISSN 1951-6851. Available from Internet: <https://doi.org/10.2991/icmt-13.2013.57>.

LUO, C.; JIN, L.; SUN, Z. Moran: A multi-object rectified attention network for scene text recognition. **Pattern Recognition**, v. 90, p. 109–118, 2019. ISSN 0031-3203. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0031320319300263>.

MONTAZZOLLI, S.; JUNG, C. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: **2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2017. p. 55–62.

PAL, S. et al. Deep learning in multi-object detection and tracking: state of the art. **Applied Intelligence**, v. 51, 09 2021.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

REDMON, J.; FARHADI, A. Yolo9000: Better, faster, stronger. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017.

REDMON, J.; FARHADI, A. **YOLOv3: An Incremental Improvement**. 2018.

RUANGSANG, W.; ARAMVITH, S. Efficient super-resolution algorithm using overlapping bicubic interpolation. In: **2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)**. [S.l.: s.n.], 2017. p. 1–2.

SHI, B. et al. Aster: An attentional scene text recognizer with flexible rectification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 41, n. 9, p. 2035–2048, 2019.

SILVA, S. M.; JUNG, C. R. License plate detection and recognition in unconstrained scenarios. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018.

WANG, C.-Y. et al. Cspnet: A new backbone that can enhance learning capability of cnn. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2020.

WANG, W. et al. Scene text image super-resolution in the wild. In: VEDALDI, A. et al. (Ed.). **Computer Vision – ECCV 2020**. Cham: Springer International Publishing, 2020. p. 650–666. ISBN 978-3-030-58607-2.

WEN, Q. et al. Aerial image object detection based on improved yolov5. In: IEEE. **2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)**. [S.l.], 2022. p. 561–564.

ZHANG, Y. et al. Residual dense network for image super-resolution. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2018.

ZHU, H. et al. Attention mechanisms in cnn-based single image super-resolution: A brief review and a new perspective. **Electronics**, v. 10, n. 10, 2021. ISSN 2079-9292. Available from Internet: <https://www.mdpi.com/2079-9292/10/10/1187>.