Automatic License Plate Recognition System for Parking Management and Access: A Computer Vision and OCR Approach

Rodrigo Freire Bastos[a,b], Hebert Ferreira Azevedo [a,b], João Pedro Almeida Miranda Silva[a,b], Thâmara Raíssa Monteiro Lins[a,b]

[a] *Senai Cimatec, Brazil*
[b] *Junior Researcher at IEEE RAS CIMATEC*

Abstract: This paper proposes an automatic license plate recognition system using computer vision techniques and Optical Character Recognition (OCR). The proposed ALPR system aims to enhance license plate identification accuracy for parking lot management and access. Based on pre-processing algorithms for noise reduction and edge detection for ROI identification, the system utilizes actual plate measurements to enhance the quality of its results. By incorporating OCR post-processing techniques, the system recognizes character and achieves promising results in character identification. The potential of this system is exposed through experimental analysis.

Keywords: license plate; optical character recognition; computer vision; parking management; region of interest.

Sistema de Reconhecimento Automático de Placas de Veículos para Gestão e Controle de Estacionamentos: Uma Abordagem de Visão Computacional e OCR

Resumo: Este artigo propõe um sistema de reconhecimento automático de placas de licença usando técnicas de visão computacional e Reconhecimento Óptico de Caracteres (OCR). O sistema de ALPR proposto tem como objetivo aprimorar a precisão na identificação das placas de licença para gestão de estacionamentos e controle de acesso. Com base em algoritmos de pré-processamento para redução de ruído e detecção de bordas visando a identificação de Regiões de Interesse (ROI), o sistema utiliza medidas reais das placas para aprimorar a qualidade de seus resultados. Ao incorporar técnicas de pós-processamento OCR, o sistema reconhece caracteres e alcança resultados promissores na identificação de caracteres. O potencial do sistema é exposto por meio de análises experimentais.

Palavras-chave: placa de carro; reconhecimento ótico de caracteres; visão por computador; gestão de estacionamento; região de interesse.

1. INTRODUCTION

ALPR is an acronym for Automatic License Plate Recognition, is a computer vision application that uses images and videos combined with vision algorithms to read, detect and extract information from License Plates [1]. The basic ALPR system involves a camera, and Image processing software such as OpenCV, TensorFlow, Pytorch, Keras. In addition, one path that can be followed is the use of Optical Character Recognition (OCR) software or libraries.

The camera is responsible for providing all the information needed to process and apply license plate recognition algorithms. While the processing software analyzes and interprets the images captured by the camera, the OCR software recognizes characters from the images, whether scanned, printed, or handwritten [2].

Nowadays the variety of applications for ALPR systems are significant, such as surveillance, parking management, tolling, law enforcement, and border control, the main area which this paper proposes is parking lot management and access control [3].

Park control and management has a huge advantage in comparison with other ALPR systems, factors such as distance plates, high number of plates, and perspective distortion are problems that the park management and control system normally will not find [3]. However, there are still hard problems to solve for this application, an example is low illumination and false ROI (region of interest) detection.

However, despite its many benefits, using ALPR for access control and parking loT (Internet of Things) management comes with a unique set of difficulties. Accurate license plate identification is severely harmed by poor lighting, which requires sophisticated algorithms and adaptive methods to successfully overcome. Furthermore, inaccurate ROI detection is quite common due the environment changes, so in a few cases, the ALPR system will not be able to detect the plate at the first time, so accordingly adaptive threshold and segmentation methods shall be used.

In recent years, many systems have gone towards the utilization of learning methods, especially neural networks including Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs), to build on sophisticated vision algorithms. While this approach has gained popularity for its impressive capabilities, this paper proposes a ALPR (Automatic License Plate Recognition) method that relies on computer vision techniques including image segmentation, contours detection and image cropping [4]. Due to the pursuit for an applicable approach, this research brings the integration of a computer vision method, with a widely used post-processing technique for character recognition, known as Optical Character Recognition (OCR) [5]. This work attempts to propose a convincing and effective answer to the complex problems raised by ALPR systems by changing the common approach, using well known computer vision (CV) methods, such as color-space modifications, edge detection by canny algorithm and some geometry analysis.

2. METHODOLOGY

There are a variety of ways to design an ALPR system. In this paper it has been proposed a classic design widely used for object extraction, going through pre-processing and post-processing techniques [3]. The usage of modern programming language with a large number of libraries and a well based character recognition method were aspects that assisted the present ALPR development.

2.1 Algorithm

2.1.1 Pre-processing

The accuracy and performance of the ALPR system is directly associated with its ability to deal with the variables presented by the environment in which the license plate is located. This includes situations where unwanted objects are in front of the license plate or the position in which the camera is located [4]. To overcome these situations, it is necessary to use a pre-processing algorithm that allows the greater uniformity and capture of the data and the better recognition of the same.

The main reason for pre-processing an image is to minimize or exclude any disturbance that can lower the system's accuracy. This paper proposes a straightforward algorithm that converts color spaces and applies edge detection via the Canny algorithm.

As the reason for pre-processing images is to reduce noise, changing image color-space follows the same idea, changing the BGR color-space to a grayscale image has many advantages, grayscale images are less susceptible to external noises, grayscale images require less memory and processing power to manipulate, also easier to work, knowing that grayscale images pixels goes between 0 to 255, only one value, different from the BGR color-space, that has a value for each color, blue, green and red (BGR). In python-OpenCV there is a pre-built function to change color-spaces, cvtcolor, that can manipulate over 150 color-spaces conversions [6].

After changing the image color-space, to reduce the image unnecessary content, a classic algorithm, developed by John F. Canny can detect edges, which helps ALPR systems to identify the plate [7]. Once all the edges have been detected, the ALPR system can extract specific ROI based on the necessity, such as the license plate, from the rest of the image. By selecting edges that are closed (its first and last vertices are connected) and have four sides, it is possible to identify where the license plate is. The search for four side polygons occurs because the license plate has a rectangular shape, as well as the shapes we seek in the image.

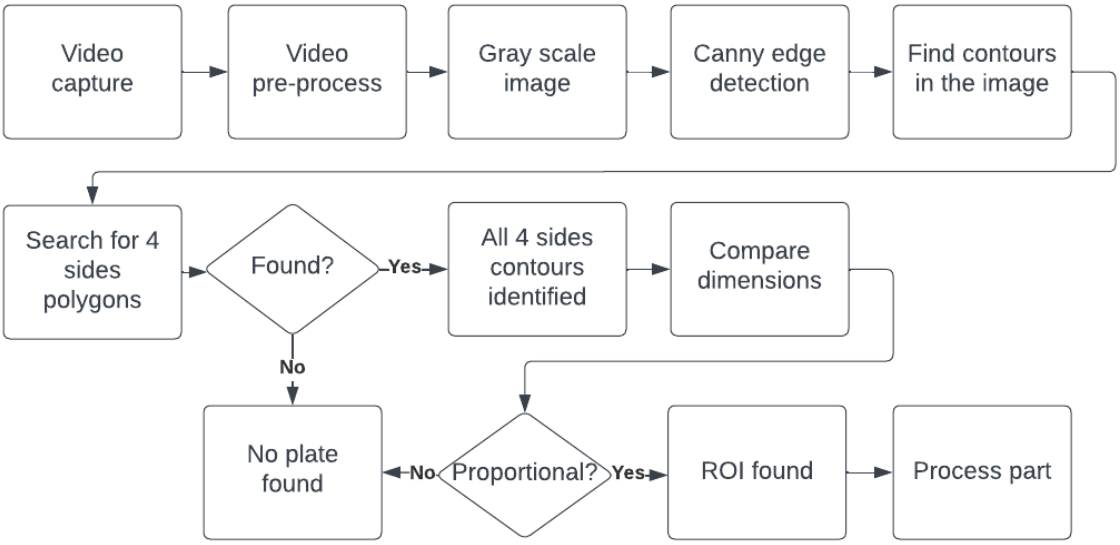
In the final step of image pre-processing, another technique to increase the algorithm accuracy is the edge proportion rate, in which is used comparing the virtual four side polygons (Sr) size rate (in pixels measured) and the real plate size rate (in centimeters measured), that can be seen in the equation below, Eq. 1.

$$Sr = \frac{height}{width} \simeq 0.325 \quad (1)$$

By dividing these measurements we have a adimensional value for the size rate, also it is possible to compare a rectangle that has the same height and width proportion compared to the real size license plate.

Then, in the image pre-process stage, we follow a series of steps to facilitate license plate recognition by reducing the noise to detect false positives. Broadly speaking, merging all these computer vision techniques is possible to select our region of interest, also known as the license plate, as observed in the following diagram, Fig. 1.

Figure 1. Pre-processing algorithm flowchart.



2.1.2 Post-processing

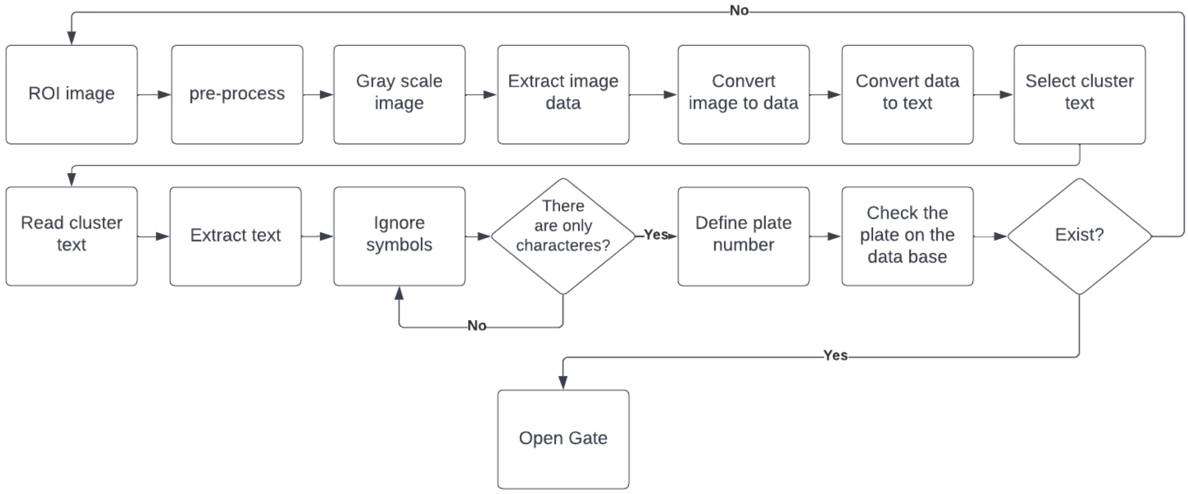
In the post-processing stage of Automatic License Plate Recognition (ALPR) using the Optical Character Recognition (OCR), we focus on refining the extracted text to ensure high accuracy and reliability. Without the last step of image pre-processing, the output may contain some errors or noise due to varying lighting conditions, image artifacts, or other challenges. To address these issues, we ensure that the correct ROI is achieved, and for better results, we convert the ROI image to grayscale, that as seen before, contains less noise than a BGR image.

In addition, one common tool to perform OCR for python is Pytesseract, a powerful open-source optical character recognition library that can extract text from images or scanned documents, enabling automatic conversion of visual content into machine-readable and editable text.

Using this tool, the system is able to gather data from the image, by selecting the type of data we want to extract, in ALPR systems is Text type, we can work on the right text format to gather, since the OCR is adept to detect various symbols, including but not limited to backslashes (), hyphens (-), underscores (_), semicolons (;), and more. The current license plate format in Brazil, the Mercosul license plate, has the following format “XXX YXYY” where “X” is a letter and “Y” is a number. Consequently, the system intends to detect 7 characters, rejecting the symbols shown above (backslashes, underscores, semicolons, and more), listing only the relevant characters to identify the plate as in the diagram below, Fig. 2.

With the license plate recognized and stored in a variable, park management and control can be applied.

Figure 2. Processing algorithm flowchart.



2.2 Optical Character Recognition (OCR)

The Optical Character Recognition is a method employed to identify, separate, and interpret characters present in images. This allows characters to be converted into data that computers can easily read and manipulate. The use of OCR technology enhances efficiency by automatically reading and processing characters present on license plates.

Various variables must be considered while implementing this technology, including the quality of the images containing the characters, camera positioning, the presence of characters with complex formats from certain languages, defects present in the license plate itself, and lighting.

Lighting is a crucial requirement in computer vision as it enhances visibility of details and improves image contrast facilitating edge detection and greater detail capture. The system's location may have insufficient illumination, which can complicate the process of extracting data from a license plate. Artificial lighting is a potential solution to this problem.

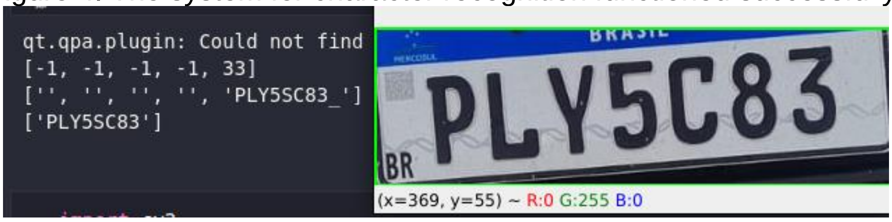
Figure 3. License plates under different lighting conditions [1].

3. RESULTS AND DISCUSSION

After using the OCR tools mentioned in this article, the results were obtained in identifying the characters on the license plates. Where, these results depend directly on some external factors to present greater accuracy and assertiveness.

In the project, the result with greater presence is shown in Fig. 4, where the character recognition system was successful in functioning and presented positive results, and from a code that analyzes the ROI of the image, it was able to identify the region of image where the plate was located and with the help of OCR tools, the program provided the list of identified characters.

Figure 4. The system for character recognition functioned successfully.

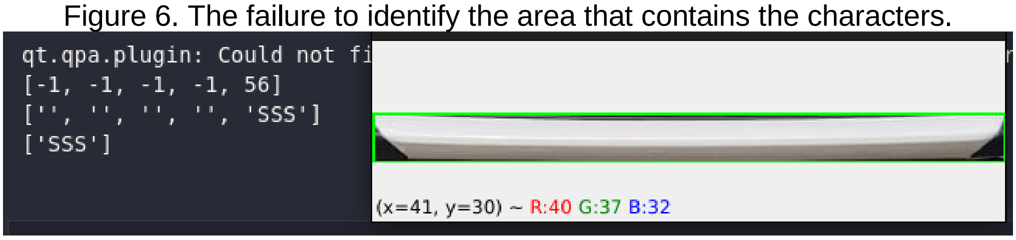


Another result was that the program presented problems regarding the recognition of the characters in the image, demonstrating difficulties in identifying the characters in the image, as shown in Fig. 5, where despite identifying the area of the image that has the characters, the code does not brought the expected result in identifying the characters in the image. This notable issue underscores the significance of lighting conditions and their influence on the efficacy of character recognition systems, warranting further investigation and potential adjustments to enhance system performance.

Figure 5. Case where the characters are not recognized.

Another malfunction in the character recognition code in an image is shown in Fig. 6, where the failure is in identifying the area that has the characters. This occurs because the system works by identifying the largest rectangular area, which in most cases in the images will be the vehicle's license plate, but in this case the program recognized another area as the largest rectangular region, which resulted in the code malfunctioning.

Figure 6. The failure to identify the area that contains the characters.



The analysis of the success of the project regarding the results was done by observing the number of times that the ROI of the image and the characters were correctly identified. In total, tests were carried out with 12 images, in different lighting conditions and angles. The experimental results used as a basis for the design performance analysis are presented below in the table.

Table 1. Experimental results

Functional Images	9
Non-Functional Images(Brightness)	2
Non-Functional Images(Angle)	1
Non-Functional Images(plate area not found)	3

4. CONCLUSION

In this article, a code system for character recognition in images was designed using the Python programming language and two of its libraries for image analysis and character recognition, OpenCV and Pytesseract, it was possible to provide a system that can be easily implemented in various areas of control using the identification of characters in images. Despite this, the focus of the project was the use of the program in a license plate recognition system that can be used in a turnstile, for example. The code managed to present positive results, having a high assertiveness for sharp images captured from the front, however in some cases of images with high luminosity or those that were captured at an angle that does not favor the operation of the OCR tools. Soon, the project reached the initial objective, in addition, the OCR area is expanding and has several tool options for character recognition, which provides the opportunity to develop this project in the long term, that said, the next steps for the development of the project, aiming at its expansion, will be the implementation of sensors and to embed the system, with the objective of using it in a concrete application with the software and the hardware.

Acknowledgments

Acknowledgements to IEEE RAS CIMATEC.

5. REFERENCES

[1] MUSTAFA, Twana; KARABATAK, Murat. Challenges in Automatic License Plate Recognition System Review. **In: 2023 11th International Symposium on Digital Forensics and Security (ISDFS).** IEEE, 2023. p. 1-6.

[2] MEMON, Jamshed et al. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). **IEEE Access**, v. 8, p. 142642-142668, 2020.

[3] LIN, Cheng-Hung; LI, Ying. A license plate recognition system for severe tilt angles using mask R-CNN. **In: 2019 International Conference on Advanced Mechatronic Systems (ICAMechS).** IEEE, 2019. p. 229-234.

[4] KIM, Taewan; KIM, Eungtae. A Vehicle License Plate Recognition System Using Morphological ROI (Region of Interest) Map Generated from Morphology Operation. **In: Journal of Physics: Conference Series.** IOP Publishing, 2017. p. 012004.

[5] GARG, Khushi et al. License Number Plate Recognition (LNPR). In: **2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)**. IEEE, 2022. p. 1261-1267.

[6]OpenCV: Changing Colorspaces. **OpenCV**. Available at: <https://docs.opencv.org/3.4/df/d9d/tutorial_py_colorspaces.html>. Accessed on: 30 jul. 2023.

[7]OpenCV: Canny Edge Detection. **OpenCV.** Available at: <https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html>. Accessed on: 30 jul. 2023.