Aman Patel

May 2, 2020

CSCI-B 456

License Plate Localization and Character Recognition

**Introduction**

License plate localization and recognition has been a staple in the image processing community since its first implementations in the late 1970s. Although the problem has been around for over 40 years, computer vision researchers are still attempting to improve the fundamental algorithm to improve runtime and correctness. Some of these improvements include accommodating for skewed license plates and improving the accuracy of optical character recognition (OCR) neural networks.

**Problem Statement**

License plate OCR is commonly used in parking garages and toll booths to accurately determine the license plate numbers of cars that violate their respective rules. While accuracies of nearly 100% have been achieved by some license plate OCR implementations, there is still room for improvement. The problem statement for this project is, “How do image segmentation and preprocessing impact the correctness of license plate OCR?

**Algorithm**

The license plate localization and character recognition algorithms have four primary steps: preprocessing, plate localization, character segmentation, and character recognition (Xie). To begin, the image undergoes a series of preprocessing techniques. The image is converted from the RGB color space to grayscale, as shown in Figure 1.



Figure 1: Conversion from RGB Color Space to Grayscale

Then, the image’s histogram is equalized to increase contrast. This transformation is shown in Figure 2.



Figure 2: Image Histogram Equalization to Increase Contrast

The final step of preprocessing is edge detection. Roberts edge detection is used because it establishes the borders of the license plate and characters well without including too much detail. The result of edge detection is shown below in Figure 3.



Figure 3: Roberts Edge Detection

 Once edge detection is complete, the license plate localization step can commence. This step begins with a vertical line erosion of 15 pixels. This removes all elements of the image that do not contain a vertical line with a length of at least 15 pixels. Performing a vertical line erosion isolates the left and right edges of the license plate from the rest of the image, as shown in Figure 4 below.

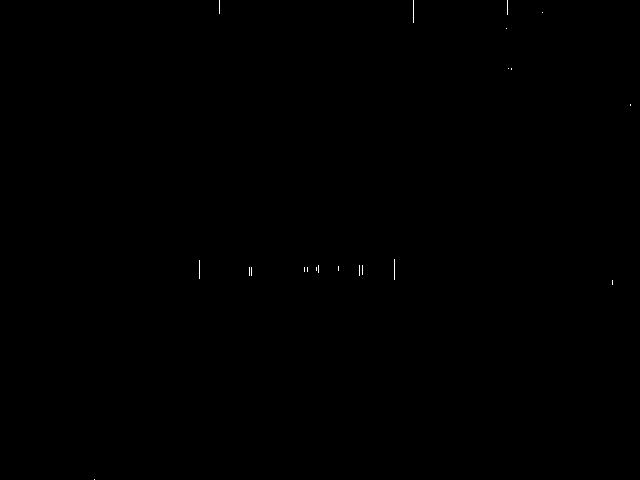


Figure 4: Vertical Line Erosion

The next step in localization is morphological closing. Morphological closing removes the gaps between elements of an image. In this case, a rectangle is used to connect the vertical lines formed via erosion. The result of the morphological closing is shown in Figure 5.

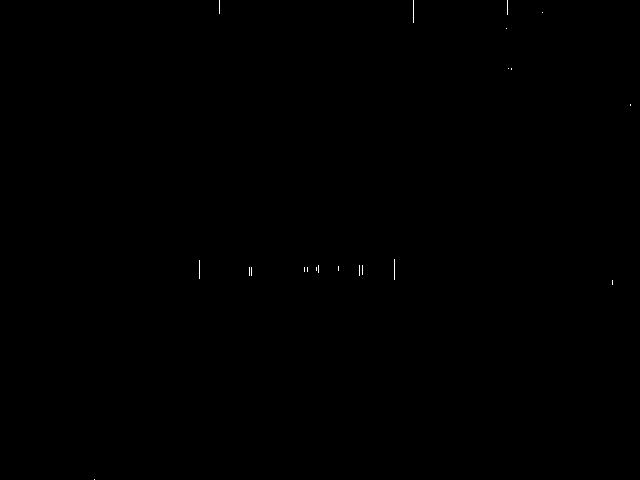
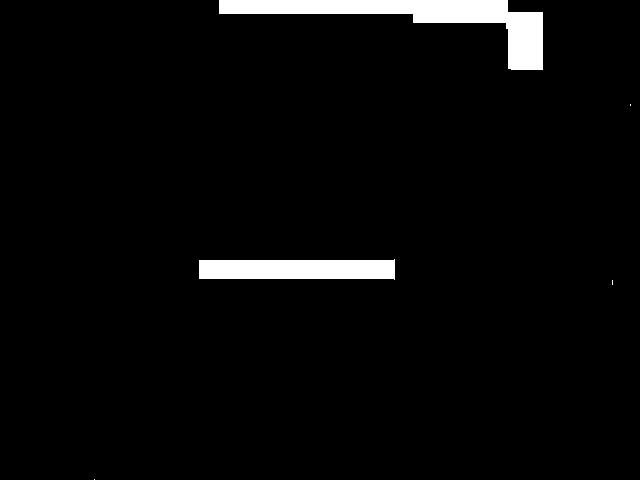


Figure 5: Rectangular Morphological Closing

Connected components are then found for the closed image, and the component with the aspect ratio closest to the license plate in question is selected. In this example, there are two regions with an acceptable aspect ratio. They are distinguished by comparing their y-positions. If one is too high, it is likely not the license plate, as license plates are typically found near the middle or bottom of an image. Once the appropriate region is determined, a corresponding rectangular mask is created, as shown in Figure 6.

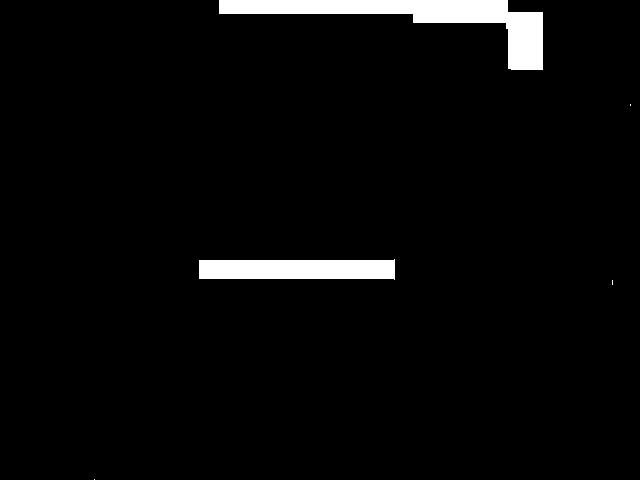
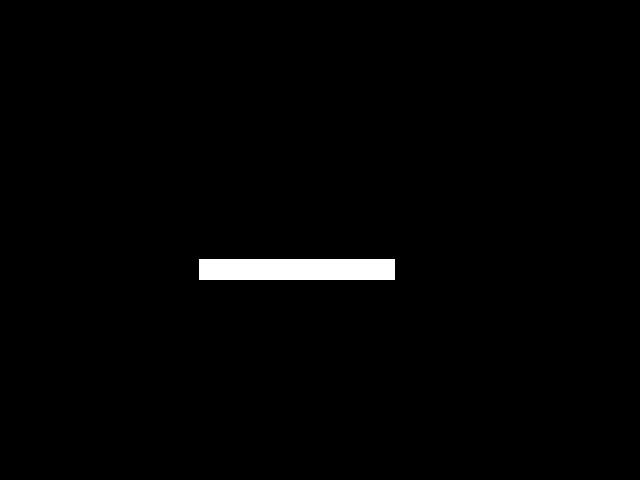


Figure 6: Selection of a Candidate Region and Creation of a Mask

The mask is then applied to the equalized grayscale image to locate the plate. The plate is then isolated from the background. This is shown below in Figure 7.

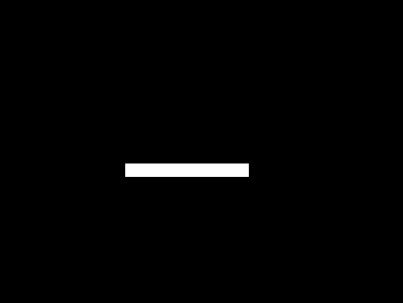






Figure 7: Application of Mask to Grayscale Image and License Plate Isolation

The plate must undergo a series of denoising techniques to isolate the text and segment into characters. First, the image is binarized with thresholding, removing cloudiness around the borders of the characters. MATLAB’s bwareaopen function is then used to remove any connected components that are smaller than 100 pixels. Any mostly white or mostly black rows are removed, as well as any fully white or black columns on the very left and right sides of the image. The bwareaopen function is run again to remove any small noise resulting from the row and column removal operations. These transformations are shown below in Figure 8.



Figure 8: Binarization, Denoising, and Isolation of Text from Equalized Plate

Connected components are found once more to determine the final coordinates of each character. A section of the image is isolated between each component’s minimum and maximum -coordinate. These images are resized to for input into the artificial neural network. Training data for the attempted neural network was downloaded from Github and included a wide variety of characters found on license plates in the United Kingdom (matthill). The downloaded box and tif files are commonly used as training inputs for a Tesseract OCR neural network. Unfortunately, after countless attempts, I was unable to get the neural network to function properly. Nevertheless, I was able to use MATLAB’s built-in OCR function to quantify the impact of preprocessing and license plate segmentation. The test images used were downloaded from the University of Zagreb and included 394 images of European license plates in various conditions (Ribaric).

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Attempted (characters) | Correct (characters) | Average Runtime (s) |
| Original | 2877 | 1058 | 0.821 |
| Segmented | 2877 | 2120 | 0.915 |
| Smoothed | 2877 | 2144 | 0.938 |

Table 1: Raw Data

As shown in Table 1, there was a significant increase in accuracy after preprocessing and segmentation. With regards to average runtime, applying MATLAB’s OCR to the original image was only slightly faster than applying OCR to the segmented image. MATLAB’s OCR functioned well in ideal conditions, where the license plate was uniformly lit, had a white background, and was not skewed. However, it was not accurate in non-optimal conditions, such as low lighting or if the license plate was rotated or skewed. Preprocessing and segmentation reduced the error in non-optimal conditions, but there were some situations where processing did not have a significant impact on accuracy. For example, the processing method outlined was not able to detect rotated plates effectively. It was also not able to detect plates with two stacked lines of text, which comprised approximately 10-12% of the testing dataset. Smoothing the borders of the characters after preprocessing and segmentation slightly improved accuracy. While it reduced the number of misclassifications by the OCR, it also distorted some characters, causing misclassification.

**Improvements**

Given my implementation resulted in an accuracy of approximately 75%, it can be greatly improved. First, different checks can be added for rotated or skewed plates. This would improve accuracy significantly, as many skewed plates resulted in the incorrect selection of plate location and incorrect character isolation. Another improvement is smoothing the borders of characters before running them through the neural network. This increases the likelihood the character will be recognized and increases the similarity between the input character and the training data. This improvement was implemented, and the impact of the improvement is shown in the bottom-most row of Table 1. A third potential improvement is utilizing more specific processing techniques to isolate the text on the plate. For example, instead of removing outermost columns based on sum, they can be removed based on estimated character locations.

**Works Cited**

1. Kocer, H E., and K K. Cevik. "Artificial neural networks based vehicle license plate recognition." *Procedia Computer Science*. 3rd ed., Elsevier, 2011, pp. 1033-37.
2. matthill, and jovargas. “Openalpr/Train-Ocr.” GitHub, 10 Oct. 2016, github.com/openalpr/train-ocr.
3. Ribaric, Slobodan, et al. “License Plate Detection, Recognition and Automated Storage Students' Project.” *Projekt "License Plates"*, University of Zagreb, [www.zemris.fer.hr/projects/LicensePlates/english/results.shtml](http://www.zemris.fer.hr/projects/LicensePlates/english/results.shtml).
4. Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. 1999. Optical Character Recognition (1st. ed.). John Wiley & Sons, Inc., USA.
5. Xie, Fei, et al. “A Robust License Plate Detection and Character Recognition Algorithm Based on a Combined Feature Extraction Model and BPNN.” *Journal of Advanced Transportation*, Hindawi, 26 Sept. 2018, www.hindawi.com/journals/jat/2018/6737314/.