Evelina Ramoskaite
Professor Furst
CSC 481
11/19/2020

License Plate Recognition System

Final Report

Abstract

License plate recognition is a useful technology with applications in traffic management, law enforcement, surveillance, and parking management systems. This project recognized license plate character values using image processing in Matlab. Rectangular bounding boxes are used to detect the license plate, then again to segment each character within the plate. Individual character values are then estimated using a template matching technique.

Introduction

License plate recognition is an image processing application that has a great deal of utility for traffic surveillance, and law enforcement. They are most commonly used by law enforcement officers, because automatic recognition allows for plate numbers to be quickly compared to important databases. This helps find records, identify stolen or suspicious cars, and check the registration status of cars. Weather conditions, noise, distance, lighting, angles, and plate obstruction are issues commonly faced with these systems, making algorithm robustness important.

Literature review

A survey of license plate recognition work found that most systems involve variations of a three-step framework involving plate extraction, segmentation, and character recognition. The techniques used for each step vary.

Plate extraction can be performed with edge statistics and morphology, color processing, block-based grey level processing, decision trees, detection with a rectangular window, and neural networks processing (Psoroulas, et al. 2008). Edge detection has yielded an accuracy rate of 96.75% within 0.2 seconds in prior research (Chen, et al. 2012.) I decided to use edge detection followed by object identification with a rectangular window because these were techniques we learned in class.

Once the plate is extracted, the characters can be analyzed. Character segmentation typically involves the same techniques used in plate extraction, just at a smaller scale. After segmentation is complete, common methods of character recognition include neural networks classifiers and template matching. Template matching has yielded an accuracy rate of 98.8% in some research (Ozbay and Ercelebi 2005).

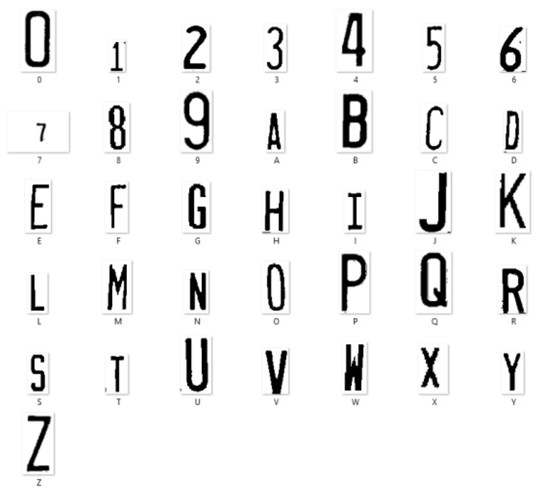
Methodology

Dataset

My dataset involved a variety of license plate photos from random American States. I obtained 25 different plates to ensure that every character, as well as a variety of states, was included.

Evelina Ramoskaite
Professor Furst
CSC 481
11/19/2020

The images were converted to binary images. The clearest and most distinctive characters were cropped out manually for template matching, input into Matlab, read, and saved as an array for use in the recognition algorithm.



```
%Template Creation
% Numbers
one = imread('0.PNG'); two = imread('2.PNG');
three = imread('3.PNG'); four = imread('4.PNG');
five = imread('5.PNG'); six = imread('6.PNG');
seven = imread('7.PNG');eight = imread('8.PNG');
nine = imread('9.PNG'); zero = imread('0.PNG');
% Letters
A = imread('A.PNG');B = imread('B.PNG');C = imread('C.PNG');
D = imread('D.PNG');E = imread('E.PNG');F = imread('F.PNG');
G = imread('G.PNG');H = imread('H.PNG');I = imread('I.PNG');
J = imread('J.PNG');K = imread('K.PNG');L = imread('L.PNG');
M = imread('M.PNG');N = imread('N.PNG');O = imread('O.PNG');
P = imread('P.PNG');Q = imread('Q.PNG');R = imread('R.PNG');
S = imread('S.PNG');T = imread('T.PNG');U = imread('U.PNG');
V = imread('V.PNG');W = imread('W.PNG');X = imread('X.PNG');
Y = imread('Y.PNG');Z = imread('Z.PNG')

number = [zero one two three four five six seven eight nine]
letter = [A B C D E F G H I J K L M N O P]
templates = [number letter]

save('templates','templates')
```

There were three steps involved in the algorithm.

Evelina Ramoskaite
Professor Furst
CSC 481
11/19/2020

1. Segmentation/ location of the license plate

My system combined edge detection with morphological transformations to prepare the image for extraction. A rectangular bounding box was then formed around the plate area.

After the image was converted to grayscale, Canny edge detection was performed. Canny edge detection was ideal because it identified the most details, but Sobel detection also produced similar results. I dilated the image and opened the image to reduce gaps in the outlines. Once the square outline of the plate was emphasized by these operations, I performed a built-in border clearing operation and used erosion to remove a lot of the surrounding noise.

Original

Canny Edge Detection

Dilated, Opened

Borders Cleared, Eroded

After these processing steps were complete, I searched for the element having the maximum rectangular value using the regionprops function in Matlab. This created the location of the bounding box representing the plate location in the larger image. This technique worked well because the images I chose were close to the car. At a greater distance, it would be prone to many errors. Once the plate was identified, the image was then cropped, preparing it for further segmentation.

Evelina Ramoskaite
Professor Furst
CSC 481
11/19/2020

2. Character segmentation

Once the license plate was successfully identified, the characters needed to be properly segmented. I used regionprops to create rectangular bounding boxes around each character in the plate. I looped through the detected objects, and only allowed the objects that had a height value greater than 1/3 and width values less than 1/3 of the image to be classified as characters. This helped to avoid objects like state names, background images, or registration stickers from being falsely identified as letters. The objects that matched this requirement were then selected for template matching.

Each Object Detected by Regionprops

```
%Reading Letters
charachters=regionprops(im,'BoundingBox','Area', 'Image');
bb = round(reshape([charachters.BoundingBox], 4, []).');
figure;
imshow(im);
for i = 1 : numel(charachters)
    rectangle('Position', bb(i,:), 'edgecolor', 'red');
end
```

3. Character recognition

Each character value was predicted using a template matching method. I calculated the normalized cross correlation of each letter relative to the templates using the normxcorr2 function in Matlab. Once cross correlations were found for each possible template, the template with the highest correlation match was declared a match.

Results and Discussion

The plate detection component of the algorithm had no performance issues. I achieved an accuracy level of 70% on a character-by-character basis. Some plate images were lower in resolution or had busier backgrounds, which likely impacted the accuracy of the character segmentation. The most frequently misread values were the O, 0, and Q. They are very visually similar characters, so this outcome was not surprising.

I initially wanted to detect the plates from greater distances but decided to limit the images used to ones that were relatively close to the car, from a central angle, and at a high resolution. When I tried the algorithm on images at a longer distance, the plate extraction portion of the system was prone to failure. This was especially true for cars with very angular designs.

For instance, the car below has a plate that is easy for a human to read, but the angle combined with the rectangular design makes it challenging to extract the correct plate area as opposed to one of the tail lights.

Conclusions

License recognition systems have a great deal of utility for important public and private applications. Image processing techniques can be used to extract plates, segment characters, and recognize their values. My approach used edge detection, morphology, and rectangular window extraction to locate the plate. It then segmented characters with object bounding boxes and calculated their correlation coefficients to identify their values.

Varying conditions, angles, visibility, and overall image quality are the challenges real-world applications of these techniques face. My attempt at this system only yielded a 70% success rate on characters that were identified in a rather controlled set of conditions.

Future work

The utility of the algorithm would be greatly improved if it were able to detect license plates from a greater distance and in real-time. This would allow it to be better used for practical applications in law enforcement and traffic management.

Applying a neural networks approach to this system could increase its robustness. Neural networks-based recognition systems tend to be more accepting of noise and varying conditions relative to template matching (Psoroulas, et al. 2008).

Evelina Ramoskaite
Professor Furst
CSC 481
11/19/2020

Works Cited

Chang, Shyang-Lih, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen. 2004. "Automatic License Plate Recognition." IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 42-53. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.2440&rep=rep1&type=pdf.

Chen, Rongbao & Luo Yunfai. 2012. An Improved Licence Plate Location Method Based on Edge Detection. Physics Procedia. 24. 1350-1356.10.1016/j.phpro.2012.02.201. https://www.researchgate.net/publication/257706849_An_Improved_License_Plate_Location_Method_Based_On_Edge_Detection.

Oz C., Ercal F. (2003) Automatic Vehicle License Plate Recognition using Artificial Neural Networks. In: Abraham A., Franke K., Köppen M. (eds) Intelligent Systems Design and Applications. Advances in Soft Computing, vol 23. Springer, Berlin, Heidelberg.

Psoroulas, Ioannis D., Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, and Eleftherios Kayafas. 2008. "License Plate Recognition From Still Images and." IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 377-391. http://pdfs.semanticscholar.org/b33e/389e2d3a43ac985df2c224dad8ef7c4e22b4.pdf.