License Plate Recognition Assignment

for Image Processing CSE2225 (2019/2020)

by Augustas Vilčinskas and Gabrielė Barteškaitė

Introduction

In this assignment we had to recognize Dutch car license plates. There were four difficulty categories. Our solution is for the First and Second categories, which means that our program recognizes yellow Dutch license plates with a single line of characters from videos taken with either a fixed or moving camera in which there is a maximum of 1 license plate per frame.

1. Localization

To extract the license plate out of each frame we apply a yellow mask on the image after converting it to hsv format and blurring. The upper and lower color values were picked by looking at the performance on training data – making sure all plates were extractable. Then we find contours (edges that form a closed shape). Canny edge detection is used for this. Then we add a rectancular bounding box on the contour with the largest bounding area. The original image is then cropped according to the rectangle we obtained. In this step we make sure that the rectangle used has dimensions similar to a license plate. The cropping is done by rotating the whole image and then cropping acording to the bounding rectangle.

Binary mask image

Contours detected in the mask

Largest mask contour on original image

Cropped rectangle

2. Segmentation

To segment the localized plate by characters we first resize the image, convert to grayscale and equalize its histogram. We apply a Gaussian blur filter to reduce noise while preserving edges. Then we applied region-based segmentation [1]. In this process we find an elevation map from Sobel edge detection, place the markers on the interest points and use them to segment the characters using the morphology watershed operation which behaves as a water shed in real life and decides what to consider as objects and what to consider as background. The thresholds were chosen to work well with the training data. This process produces a binary image on which we apply closing. We do a bit more erosion than dilation to make letters thinner. Then we count how many objects could potentialy be characters by looking at the dimensions of their slices. While doing this we look at the distaces between potential characters and group them according to where there was a bigger gap between characters. This helps us identify the possible license plate format [2] which we later use in recognition. We only proceed to the next step if eight potential character objects are obtained. We evaluated this process by looking at our results. Out of other methods we tried, this chosen path of segmentation performed best at finding potential character objects.

Correctly segmented character

3. Recognition

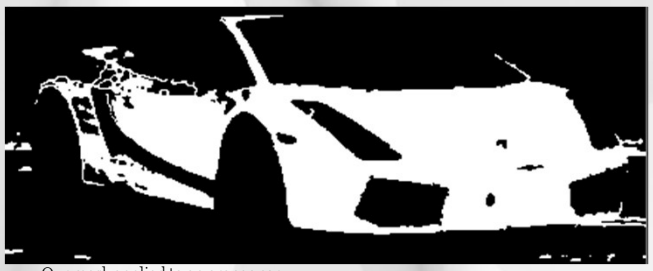
We calculate the Mean Squared Error for each of the segmented characters by comparing them with characters provided by the teaching staff. To improve our recognition, we hardcoded all the possible letter-digit ordering in Dutch license plates [2]. We find the one that corresponds to the plate we are recognizing by checking the position and type (letter or digit) of the character recognized with the lowest error and using the grouping of objects provided by our segmentation method. This helps us minimize errors where letters and digits are similar to each other. For example, in the case of S and 5. This approach performed well on our training data.

4. Majority voting

After we recognize the license plates, we perform majority voting and append its output to our result file. For this we take 3 recognition outputs. This number decreases the chances of ties. Before voting we check if all results belong to the same category (same letter-digit combination). In case we have 2 results for the plate, we take the previos majority vote output as the 3rd. In case we have 1 result, we discard it. In the cases where the only results for the specific plate are not enough to perform a vote (there is just 1 or 2) we append them to our result file as they are. By performing majority voting this way we achieved best results.

Things to improve

Our final implementation still has its shortcomings. Since we chose a static mask that is predefined, it could not work when the lightning is off or if the car color is some shade of yellow, to tackle this we could first detect the license plate object, observe the colors that are present in it and use it for the mask. This way the masking part would work much better and a system would be more adaptive. Moreover, it would also work for license plates that are of a different color. Another one of our shortcomings is the fact that 2 license plates in the same frame are rarely detected, this is because our implementation of tracking license plates was meant for one plate and we could not find another working solution for two different plates. They also could not be efficiently tracked because our mask generated some false positives for some of the frames which in turn complicated it quite a lot.



Our mask applied to an orange car

Sources

1. Comparing edge-based segmentation and region-based segmentation. (n.d.). Retrieved December 29, 2019, from https://scikit-image.org/docs/0.12.x/auto_examples/xx_applications/plot_coins_segmentation.html.
2. Vehicle registration plates of the Netherlands. (n.d.). Retrieved December 31, 2019, from https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_the_Netherlands.

TU Delft
Delft University of Technology