License Plate Detection

Beeldverwerking TI2715-B

Phil Heijkoop & Jorick Spitzen

pheijkoop - 1311115 and jspitzen - 1509373

TU Delft Delft University of Technology

Abstract

This poster details the workings of our license plate detection and reading system. Our system works in the following manner:

1. The video file is selected in the GUI
2. The video file is loaded into MATLAB
3. The frames are extracted from the video to be processed
4. The license plate is selected by contrast and intensity (by converting it to a HSI image first).
5. The area around the license plate is highlighted with a bounding box
6. The license plate area detected by the HSI contrast method is used as a mask to remove just the license plate for processing
7. The license plate is rotated where necessary, before being passed to the identification function
8. The plate identification function converts the image to a black and white image.
9. The six largest objects in this image are chosen and everything outside of this is cropped out
10. These six largest objects are then compared to our letter database before returning a string of six characters
11. The detected license plate is added to the listbox if it doesn't yet exist in there.

Introduction

We started by creating a GUI, electing to create a simple design.

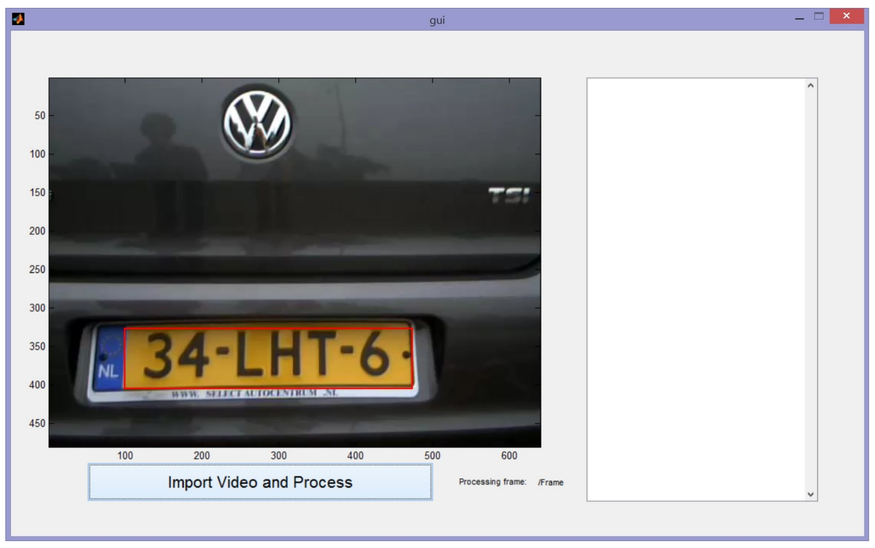

Figure 1: The GUI after detecting a plate.

We have elected to take a step-wise approach to the problem. First we attempt to segment out the license plate. Then we clean up this image to become a binary image of much smaller resolution (or multiple images if we detect multiple plates). Then we pass this image to our character recognition function. This function returns a string of six characters. Using the invariants of Dutch license plates we added the dashes, before placing the string plus other requested data into out listbox.

We later added a new axis that shows the output of the segmentation, originally for debugging reasons to see where a false positive or false negative originates, we added it now as a feature.

License Plate Segmentation

We first set about sampling several frames from the video and letting several filters run on them to see which one would yield a satisfactory discriminating result. Some results were encouraging in one frame, but yielded useless results in another (such as filtering over a single RGB channel threshold). We also attempted to use a blank license plate to detect the features in it as a reference and try to plot corresponding features in the frames[1], this failed because the blank license plate didn't have enough features to discriminate.
We then noticed the contrast value for yellow is fairly narrow, so we implemented a filter that looked for values in that range, after first converting RGB to HSI, tweaking it where needed.
This area is used as a mask to segment out the license plate from the rest of the scene.

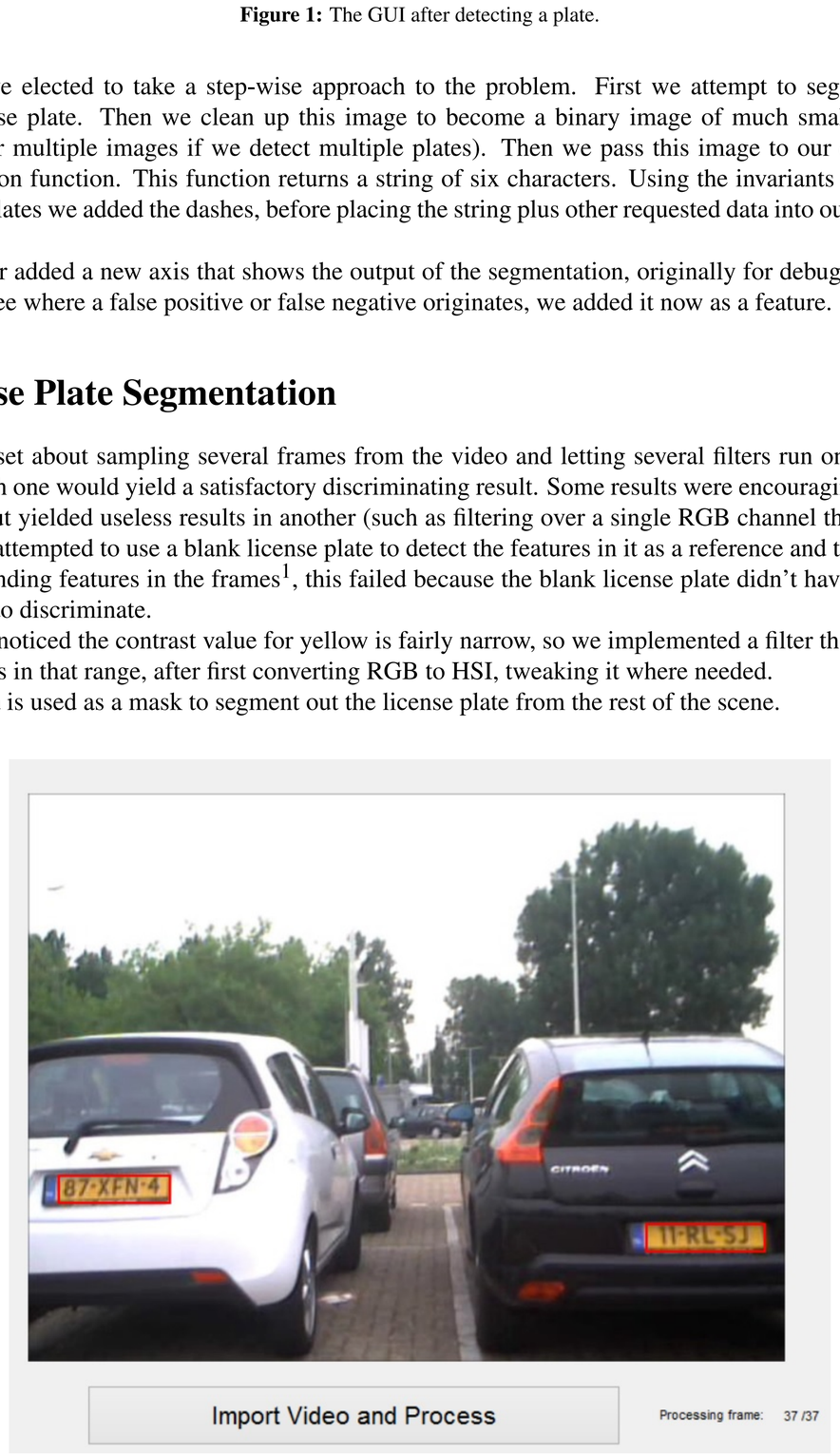

Figure 2: Demonstration of multiple license plate recognition

The segmentation has been tweaked to now not only identify multiple license plates in a scene and return them, but it can also rotate the images that have been rotated in plane.

[1] http://www.mathworks.nl/help/vision/gs/object-detection-and-tracking.html

License Plate Identification

The license plate identification of our program takes as input the cropped region within the bounding box. It filters this based on the Red RGB channel to get a black and white image.

Figure 3: The result of license plate identification.

We know that all license plates start and end with a letter or number. This allows us to perform a series of operations to identify the characters and only the characters. We take the six largest objects in the frame and send each to our letter recognition function.
In order to handle the variation in letter size and lighting we threshold based on values that depend on each scene. For each license plate we calculate that a minimum percentage of the scene needs to be populated for it to be considered a valid plate, we do this to ensure no false positives when a yellow car panel is accidentally identified as a license plate.

Letter Recognition

Given the license plate, we take the six largest areas and compare each one through our letter database. We use the fact that all Dutch license plates have the same font to create a database. We remove the vowels since these are considered 'special'. Their reintroduction is trivial, should the need arise).
A function uses regular expressions to place the dashes after the six characters have been identified. This gives an extra control layer that the license plate has a correct format.
We scale each letter from our database to compare it to our test character, many of our characters have a similar size.

We originally wanted to compare the skeleton of each letter to a skeleton database, however due to slight variations in letter size and orientation, the first test plates showed wild divergence. We are now using a straight up comparison to a scaled black and white sample of 60x40.

Now we find the error by taking the absolute value of the image subtracted by the template. We then define a threshold (which is a percentage of the total pixels in the image) under which a letter has been declared 'found' and then return the letter.

Areas for improvement and limitations

There are several small areas that could speed up the computation or increase robustness. We consider them non-essential and left their implementation for 'if we had time at the end'.

- An implementation for recognizing multiple plates in a scene requires some more time, but the implementation is fairly trivial w.r.t. what has already been done
- An area for future improvement would be selecting the license plate in a color-independent way so as to be able to handle category 4/foreign plates.
- We have removed the letter instances 'a', 'e', 'i', 'o', 'u' and the dash from the database. Adding these again when the program needs it is trivial.
- If foreign plates use a different font, this should be added as a separate database for use.
- We have created a special case identification for the number 1, since our current comparison method doesn't handle different aspect ratios very robustly. This is an area that could benefit from some added attention.
- Some segmentation results aren't as 'clean' as others and as such the letter recognition portion will sometimes mix up similar characters (ex. 5 vs 8), we fixed this by only accepting a plate if it has been recognized multiple times, however a better solution would be in a cleaner segmentation result.
- License plate recognition would benefit from less sensitivity to shaded (overhang from bumpers) portions.

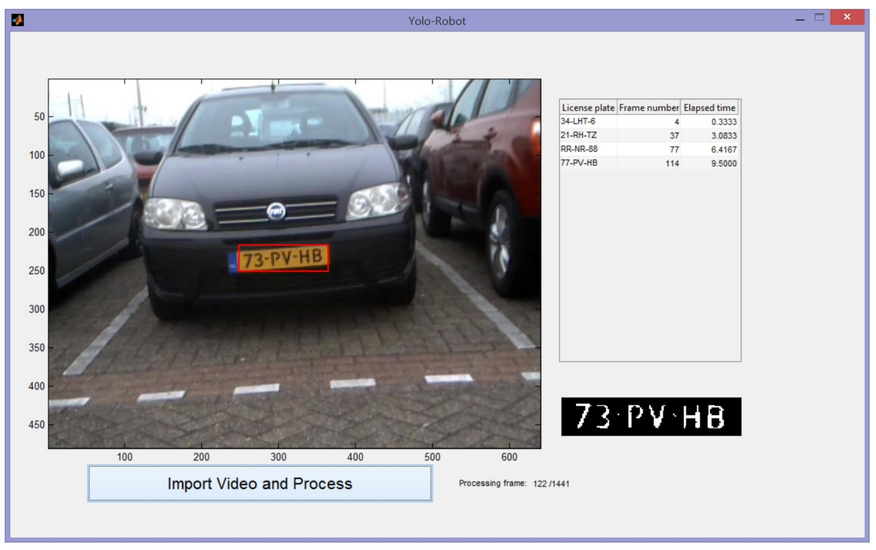

Figure 4: Example of our program running

Conclusions

Our model searches for license plates based on their color. We have tested several techniques for image segmentation and character recognition, forming a basic understanding of their uses and limitations. The program has several checks to ensure false positive values for the license plate aren't returned.