18-794 Pattern Recognition Theory - License Plate Recognition

Carnegie Mellon University

# Results

Alexandre Alahi, Alok Menghrajani

May, 2003

## 1 Discarded images

Our test database consists of 40 images. We discarded 5 images for being foreign cars (or very bad quality images). 8 images were of the front plate, which we didn't user for the character localization. 1 image had a missing part (cut seven) - we were still able to recognize it successfully.

## 2 License plate localization

We didn't get good results on the license plate localization part. Our algorithm succeeded only 3 times out of 35 (8.5%) (whereas it worked 11 out of 27 times (41%) with the training data). The reason is probably because the test set has more white cars and more light reflections.

## 3 Character localization

The character localization worked very well. For the digits, it failed only 2 times out of 193 (99% of success) (it selected extra digits due to spots on the plate). It failed to locate the letters 3 times out of 26 images (94% of successful letter selection). It was always due to a bad manual selection of the plate. If we reselect them, we are able to get the characters right.

## 4 Character recognition

We also got very good results on the character recognition part. For the digits, it failed 2 times (99% of success). The character recognition failed once (98%) (on a bad quality image).

| | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 22 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 27 | 0 | 1 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 |

Our very good results on the letter recognition part were achieved by the use of the valid state matrix. Without this matrix, we would have got less than 50% of correct results.

## 5 Computational complexity

To complete the report's section on computational complexity, we did some more tests using the profiler. The machine we are working on is a 1.3Ghz Celeron. We got similar results with the test data (4 seconds to locate the plate, 1 second to find the characters and 2 seconds to recognize all the digits). In this evaluation, we did not take into account the time needed to decompress the images and transmit or save the results.

We noticed a huge (up to a factor of 1000 between a 'matlab' style of programming and a more traditional 'C' like style). The reason is because loops and other programming constructs are interpreted, whereas the built-in matrix operations (which can often be used to replace loops) are compiled code. Now that we are aware of this difference, we could improve our code to run faster.

From a memory requirement point of view, we are storing little information (the filters for the digits and letters, which are 32x40 pixels, take about 35 KB).

To conclude, we can easily run our code in real time (since cars are usually seperated by 2-3 secondes at least).