License Plate Recognition

Group: 9

Tomas Heinsohn Huala	4326318
Remi van der Laan	4326156

This Iteration

We're closing in on the final deadline, so for this iteration we focused on improving our program resulting in a better license plate detector with better results instead of adding new features. This week we managed to improve several things. This is a summary of our improvements:

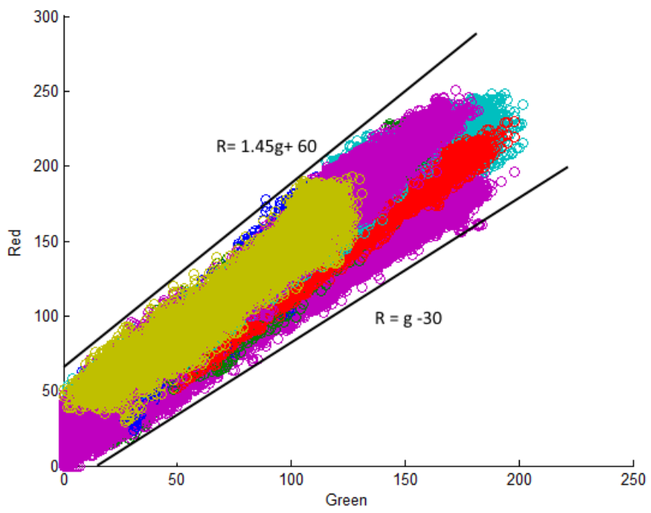
- We improved the detection of where the license plates are located.

- We changed the output of the recognized license plates so we get fewer and more accurate results in the table.

- We changed the amount of frames that are skipped to automatically get 1.3 times the total time of the video.

Plate Detection

We wanted to detect as many plate as possible, but this wasn't the case before this week. So we had to change the code which was seeking for the yellow license plates. Just like the last time we used graphs to determine what to select . To make the graph we now used the license plates which were brighter or darker than the normal license plates. This is how our new graph looks like with the red channel on the Y-axis and the green channel on the X -axis



We also said that the green and the red channel must be bigger than twice the blue channel. This is because of the yellow color which especially consist of green and red with a low value of blue. After all this the detector is able to detect the license plates of almost all of the license plates

Text Output

We improved the accuracy of the license plates shown in the table by counting which license plates are recognized more frequently. If the difference between a new recognized license plate has 3 or more different characters, then we assume it's a new license plate.

Also, we first recognized the dashes by looking for small objects, but now we add them where the two largest gaps between characters are.

Here is how our table used to look and how it looks now:

	Licence plate	Frame	Time (sec.)
1	XT-LX-6-9J	8	3.7787e-05
2	X-T-LX-69J	15	0.2530
3	X-T-LX69J	22	0.5179
4	X-T-LX6-9J	29	0.7822
5	X-T-LX-69	36	1.0484

	Licence plate	Frame	Time (sec.)
1	XT-LX-69	3	7.8411e-05
2	FP-NL-93	38	4.4119
3	63-HK-HD	77	8.5717
4	01-XJ-ND	110	10.7923
5	96-N0-JB	151	16.0984

Because of this, we were also able to compare our result with the answers of the training video:

```
RESULTS:
                    Category I     Category II    Category III   Category IV
  True positives (TP)  16 ( 53.33%)   4 ( 40.00%)   0 (  0.00%)   0 (  0.00%)
 False positives (FP)   7              7             0             0
 False negatives (FN)   7 ( 23.33%)   0 (  0.00%)   10 (100.00%)   5 (100.00%)
                    ------------------------------------------------------------
               Score   0.53           0.36          0.00          0.00

FINAL RESULT:
                    Category I+II  Total
  True positives (TP)  20 ( 48.78%)  20 ( 36.36%)
 False positives (FP)  14            14
 False negatives (FN)   7 ( 17.07%)  22 ( 40.00%)
                    --------------------------
               Score   0.49          0.36
```

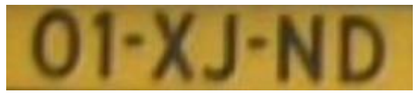
We lost a lot of points because characters like '8' and 'B' are often mistaken for each other, which causes a lot of false positives and negatives. We are currently trying to solve this by recognizing which sidecode the license plate is using, and then converting these similar looking characters to the correct one.

Other Improvements

We've now automated the amount of frames that are skipped so our processing time does not exceed 1.3 times the time of the video. The amount of frames that is skipped mostly fluctuates between 1 and 5.

We're now also using sharpening to improve the separation of the license plate with the characters, which helped to improve the accuracy of the character recognition. This is how the original and the sharpened image looks:

Without sharpening:



With sharpening:

Next Iteration

This will be the last iteration till the final deadline. Now we want to fine tune our program so we are able to get enough good results from the license plate detector.

These are the problems which we plan to fix in the next couple of days:

- An import problem is the fact that our system still has problems with detecting some of the license plates, with bright yellow or white cars for example.

- We also want to change the fact that in the results it should not be possible to place a digit next to a letter and vice versa. We've starting implementing this with the help of the standard sidecodes, but we haven't finished implementing it yet.

Lastly, we also want to spend a good amount of time on creating the last poster, because this should give anyone a good understanding of how our product works and which problems we had to overcome.

TU Delft Delft University of Technology