

License plate localization

Matej Havelka and Patrik Barták

Strategy

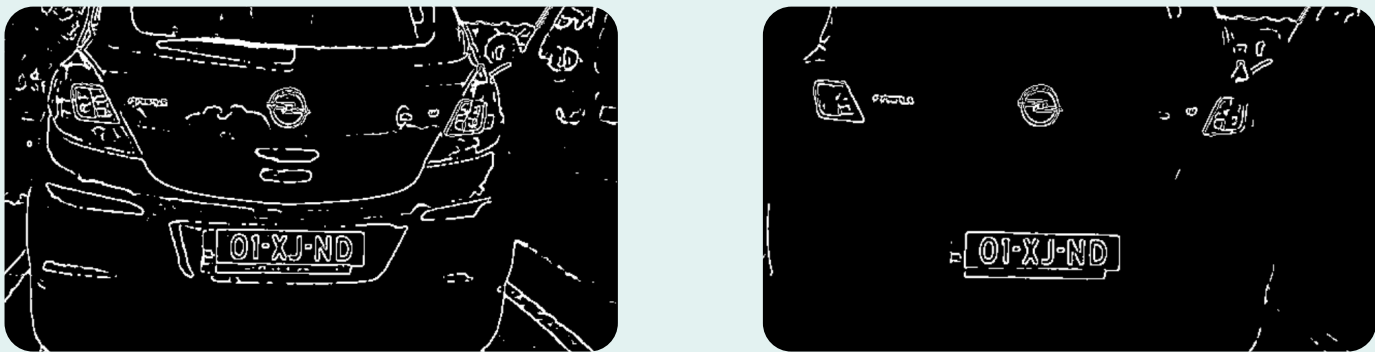
In order to maximize the number of license plates our program could localize, we created a localization strategy consisting of a preprocessing stage, two bounding methods, and a final cropping and adjusting stage. The two bounding methods consist of a main method that performs the majority of the bounding, and a backup method in case the first fails to localize any plates in the image. This consists of colour thresholding and then bounding.

Preprocessing

The preprocessing stage occurs in order to reduce noise in the image and extract the relevant edges. We did this by applying the Canny algorithm as described below:

1. A Gaussian filter is applied to reduce noise
2. The gradient magnitude and direction are calculated
3. The edges are thinned using non-maximum suppression
4. We apply a threshold and edge running to extract the most relevant edges in the image

Examples of resulting edge images, one with looser interval, one with stricter interval, are shown below.



Stage 1 – Bounds and Contours

With the relevant edges extracted, we now need to find if and where license plates occur in the image. We find the contours in the edge image using OpenCV and keep the 20 contours with the highest area, in order to remove very small objects from consideration. We then approximate each contour to a polygon and check if its properties match the license plate properties we had defined. These are described in detail on the right. If they are found, we return these contours as a mask and perform cropping and adjusting.



Evaluation

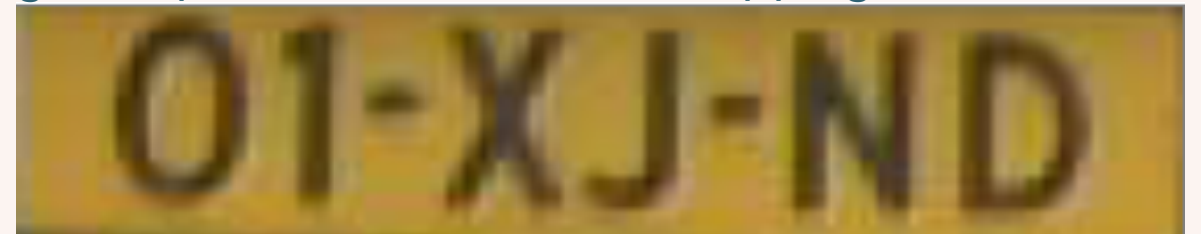
To evaluate our algorithm we have chosen to use Accuracy, specifically the ratio of found license plates over the total amount of license plates to find. We decided that a plate is "found" if it is recognized in at least one frame of the video where it is present. We have successfully managed to find each plate in category 1 and 2. However in category 3 we go lower to roughly 50% of the plates found and only about 20% are found for category 4. We have evaluated this manually based on the ground truth and the training video provided.

Stage 2 – Colour and bounds

The backup phase of our strategy occurs when no object originating from the edge image fits the the license plate properties. Given that we want to prioritize Dutch license plates, we reduce the amount of irrelevant information in the image by thresholding the colour yellow and creating a mask. This is further described in the "colour localization" section. This image is then used as the input for a second pass of the bounding and contouring process.

Stage 3 – Cropping and adjusting

After we have successfully found the license plates in the image, we need to extract them and adjust them. For each found plate we find the bounding rectangle and crop the image so it only contains the rectangle. If a license plate is tilted, our method does not remove the environment surrounding the plate, however we are working to implement the rotation cropping as well.



License place colour localization

To extract all the yellow license plates from an image. We threshold the image based on the interval of the colour yellow, and create a mask that incorporates all yellow pixels. Sometimes there might be little spikes of yellow somewhere, we solve this by opening and closing. The main issue with this method is that everything yellow from the image will be extracted, not only the license plates. This is why we then use this as input into the bounding and contour process



License plate properties

After we have successfully found our contours, we check which found object is actually the license plate. We do this by checking certain properties for all found objects. We have determined the following properties of a license plate:

1. It is created by exactly 4 points
2. The angles between the sides are roughly 90 degrees (varies based on tilt)
3. The ratio between sides is roughly 5:1 (varies based on tilt)
4. It has some minimum area, this is mainly to eliminate small rectangles that fall within the epsilon

If the found object has all of these properties we store it as a license plate. You can see that these properties yield normal looking plates.

