VEHICLE LICENSE PLATES DETECTION FROM COLOR IMAGES

Image Processing Project

REQUIREMENTS

COLOR IMAGES OF VEHICLES (AT DIFFERENT DISTANCES) WITH VISIBLE LICENSE PLATES ARE GIVEN (THE LICENSE PLATES CAN BE A LITTLE BIT ROTATED TO THE HORIZONTAL DIRECTION);

YOU'LL IMPLEMENT AN ALGORITHM TO IDENTIFY AND MARK THE AREAS WHERE THE LICENSE PLATES ARE FOUND.

BASIC STEPS OF THE APPROACH

1. Image acquiring

2. Grayscale conversion

3. Noise reduction

4. Image thresholding

Canny Edge detection algorithm

Connected Component Labeling (with/without dilatation)

Choosing the best candidate

STEP 1&2:
IMAGE ACQUISITION & GRAYSCALE CONVERSION

Loading the image should be the easiest part, because of the functionalities offered by OpenCV.

The color image will be converted to grayscale with the method applied in one of the first IP laboratories.

Big advantage of grayscale images is that is saves up a lot of memory and it is easier to work with them.

STEP 3&4:
NOISE REDUCTION AND THRESHOLDING

Reduce the noise present in the image using some gaussian filters (mathematical properties)

- Reduce unwanted information which deteriorates image quality.
- Apply Gaussian convolution $I_D = G * I_S$ - implementation may vary

Apply thresholding process to obtain a binary image (black and white image suitable for Canny algorithm)

STEP 5: CANNY EDGE DETECTION

The steps of the Canny edge detection method are given below

1. Noise filtering through a Gaussian Kernel
2. Computing the gradient's module and direction
3. Non-maxima suppression of the gradient's module
4. Edge linking through adaptive hysteresis thresholding

CANNY EDGE DETECTION ALGORITHM

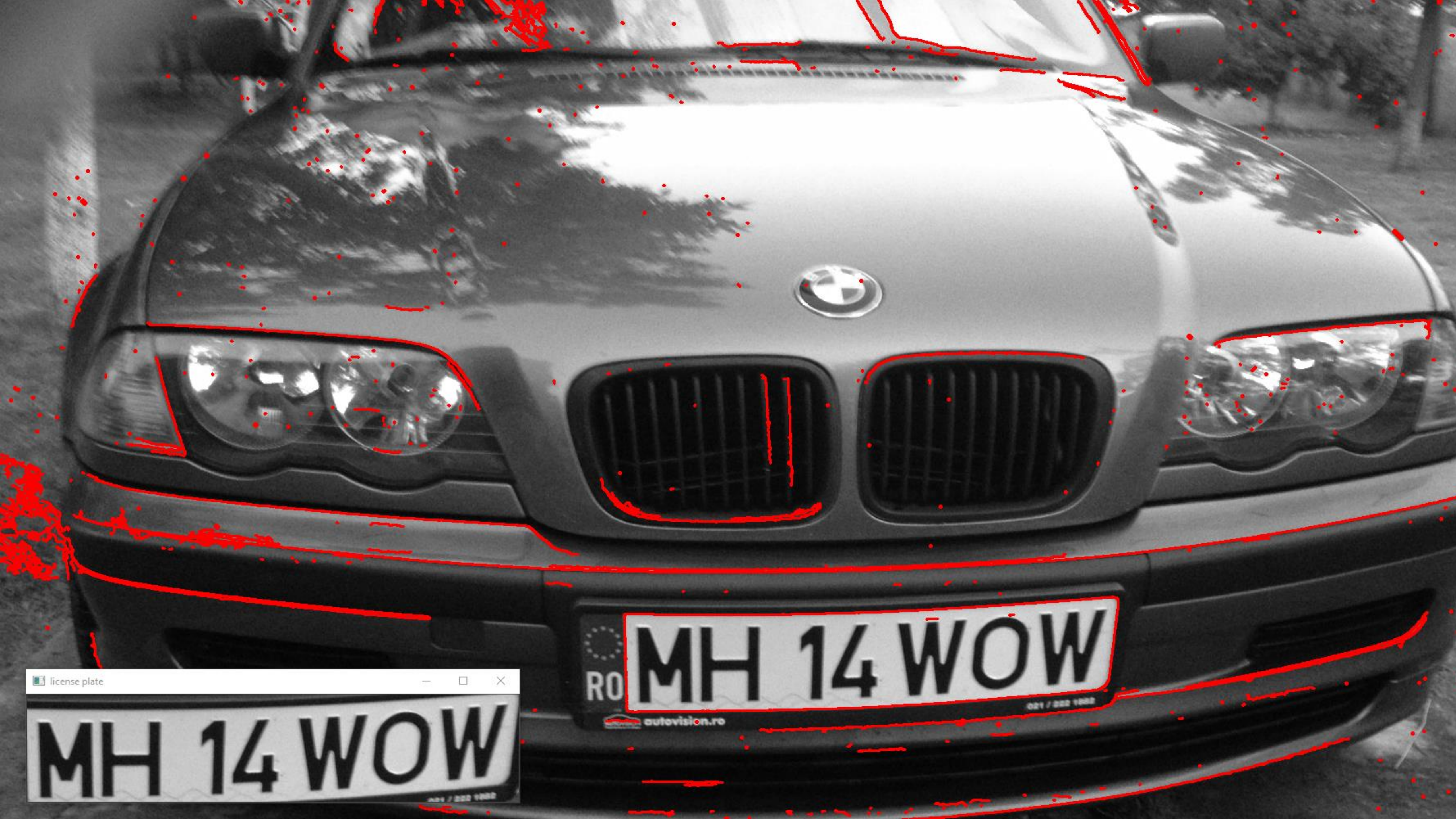
STEP 6&7: CONNECTED COMPONENT LABELLING AND FINAL DECISION

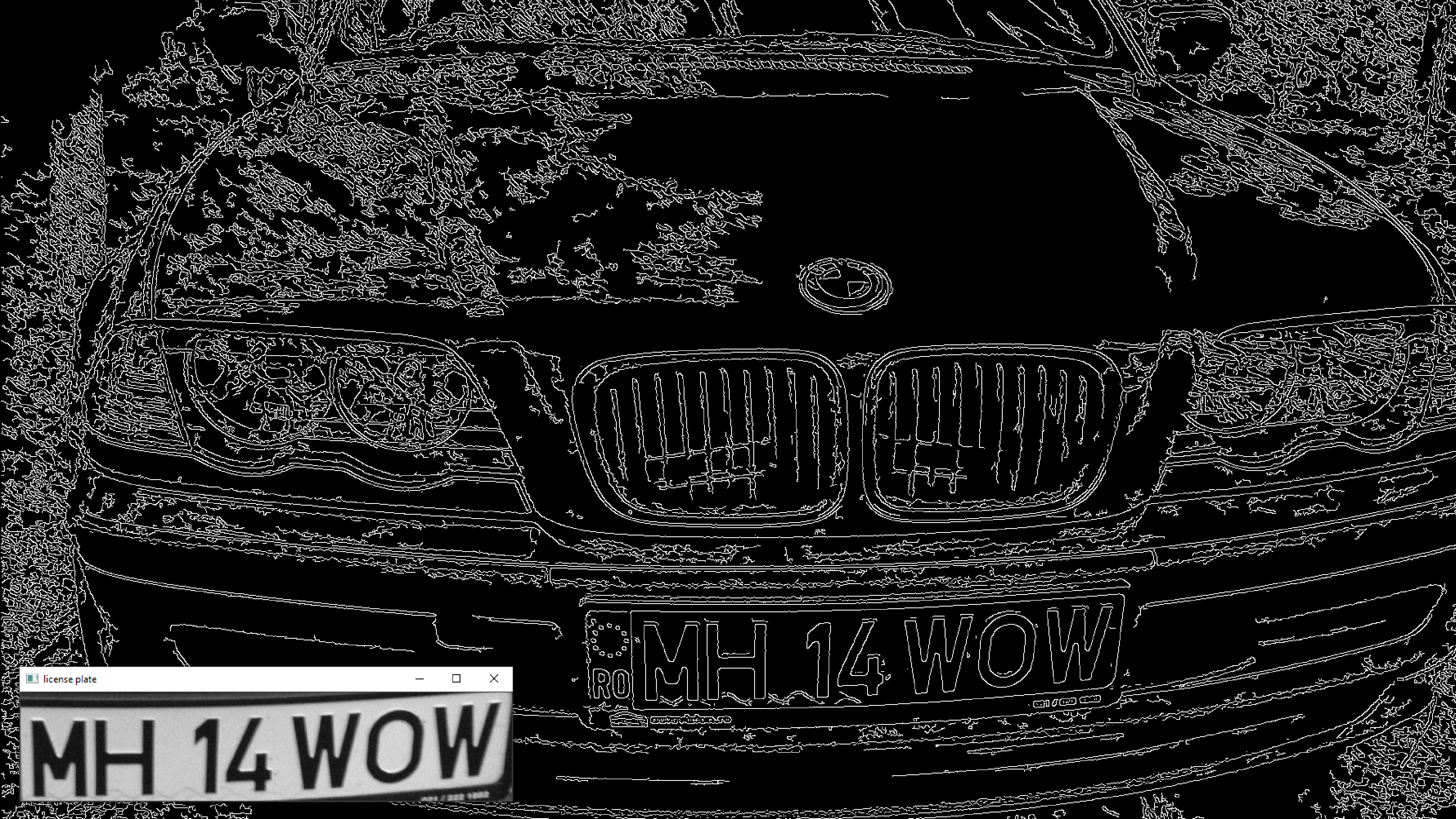
Label the same neighbouring pixels that belong to the same object.

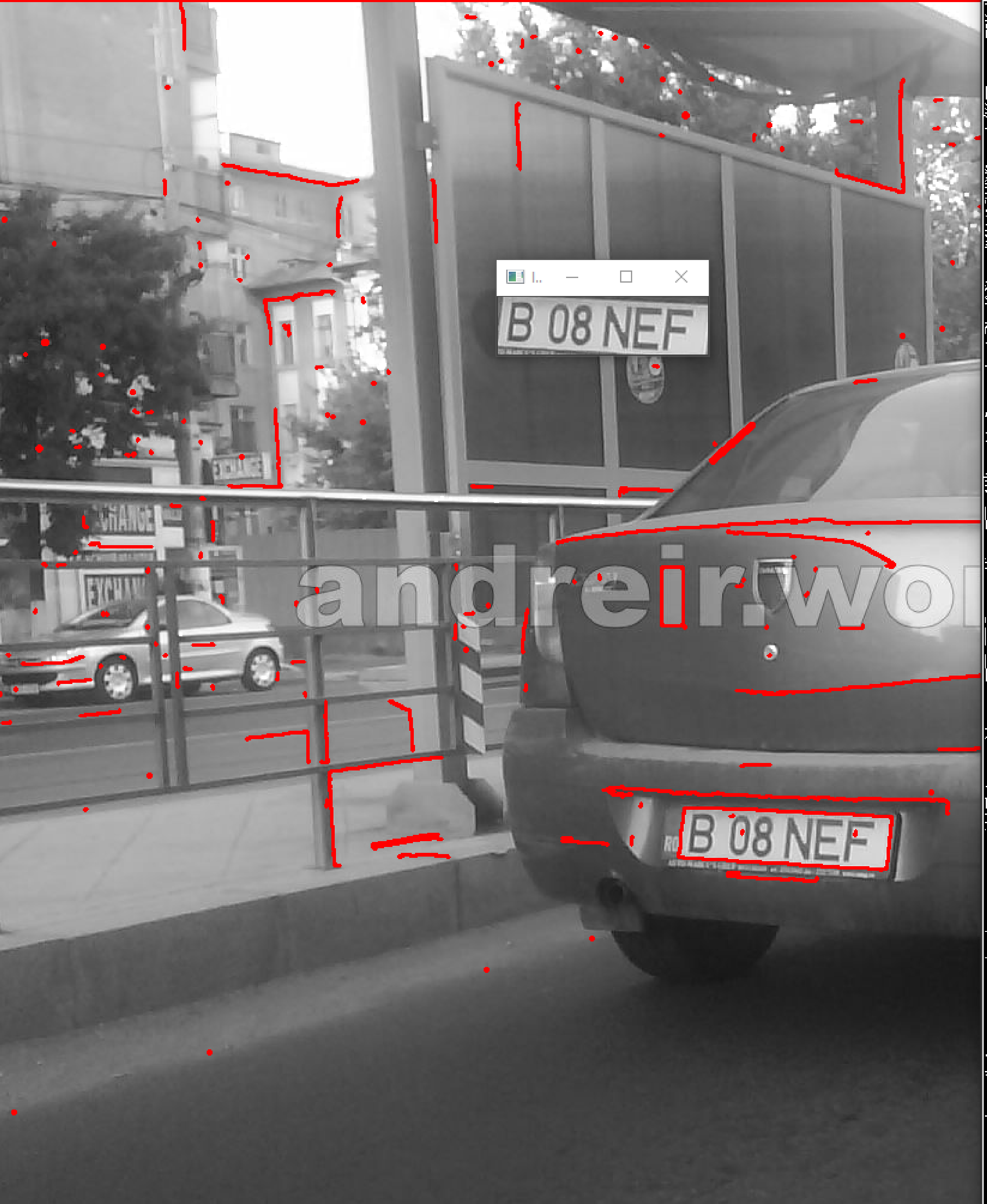
- for this process a binary image is needed (output of the Canny algorithm)

Finally, from those components, select the suitable one based on some mathematical and logical principles and rules.

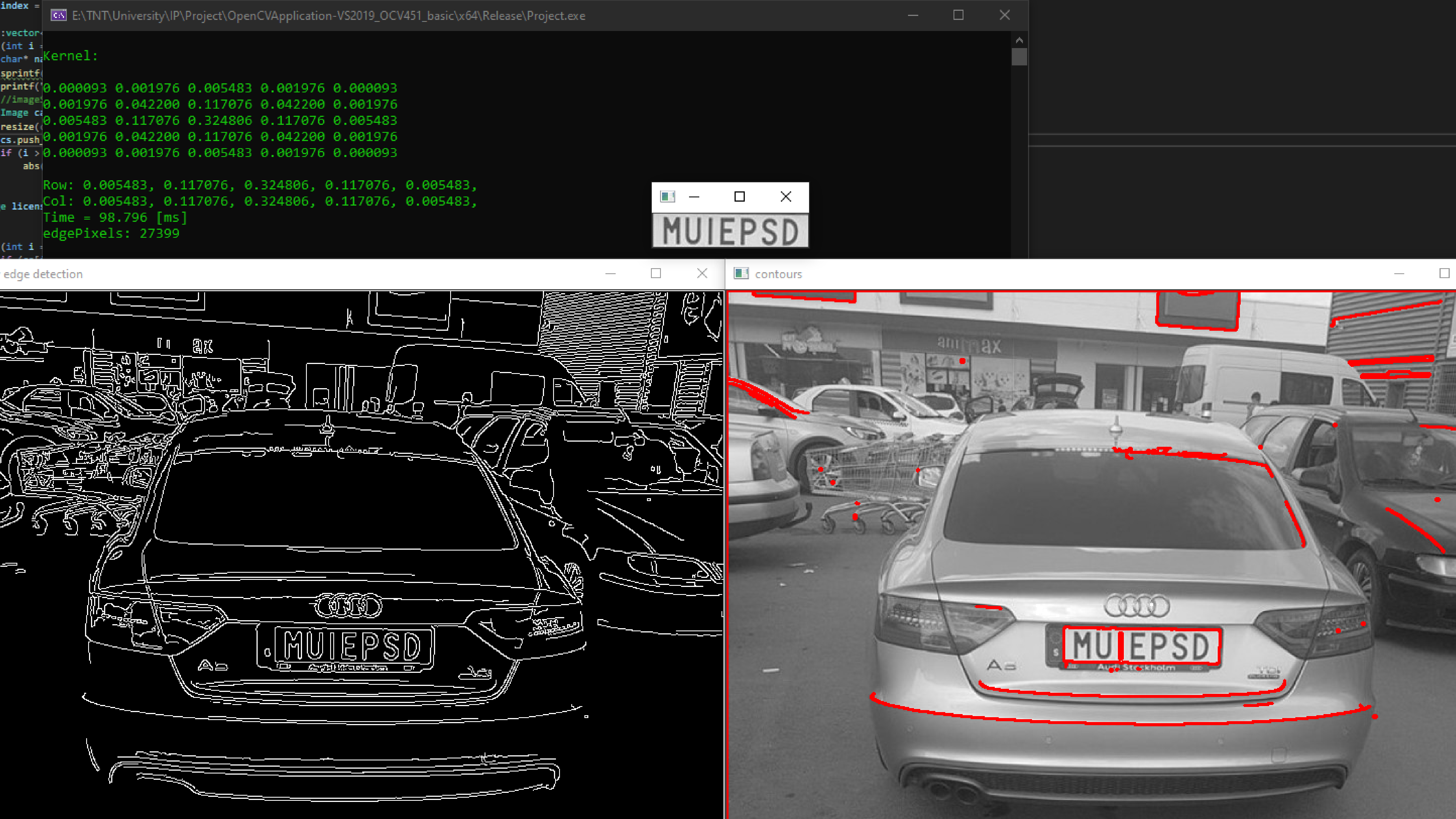
ALGORITHM PRESENTATION


MH 14 WOW
RO
021 / 222 1882
autovision.ro
license plate
MH 14 WOW
021 / 222 1882


license plate
MH 14 WOW
RO
MH 14 WOW


B 08 NEF
B 08 NEF
andreir.wor

andreir.word
B 08 NEF


E:\TNT\University\IP\Project\OpenCVApplication-VS2019_OCV451_basic\x64\Release\Project.exe
MUIEPSD
edge detection
contours

```
Kernel:

0.000093 0.001976 0.005483 0.001976 0.000093
0.001976 0.042200 0.117076 0.042200 0.001976
0.005483 0.117076 0.324806 0.117076 0.005483
0.001976 0.042200 0.117076 0.042200 0.001976
0.000093 0.001976 0.005483 0.001976 0.000093

Row: 0.005483, 0.117076, 0.324806, 0.117076, 0.005483,
Col: 0.005483, 0.117076, 0.324806, 0.117076, 0.005483,
Time = 98.796 [ms]
edgePixels: 27399
```

BIBLIOGRAPHY

http://www.iosrjournals.org/iosr-jeee/Papers/Vol12%20Issue%201/Version-2/A1201020106.pdf

https://sod.pixlab.io/articles/license-plate-detection.html

OpenCV functions – some tried out with applied functions