



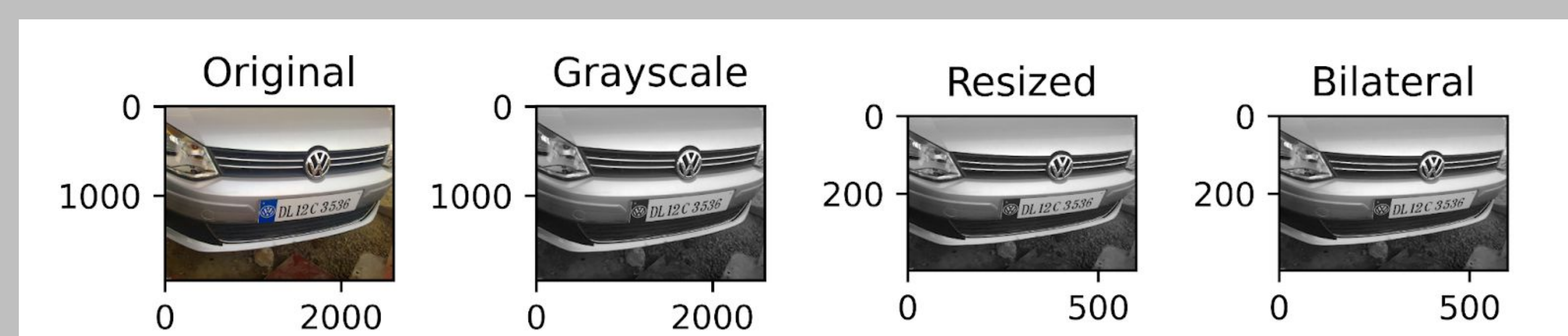
Introduction

In this poster, we will show the steps for locating and identifying car license plates from an image based on some computer vision techniques. And in this report, we will introduce methods, performances and future improvement of our program.

Steps for license plate recognition

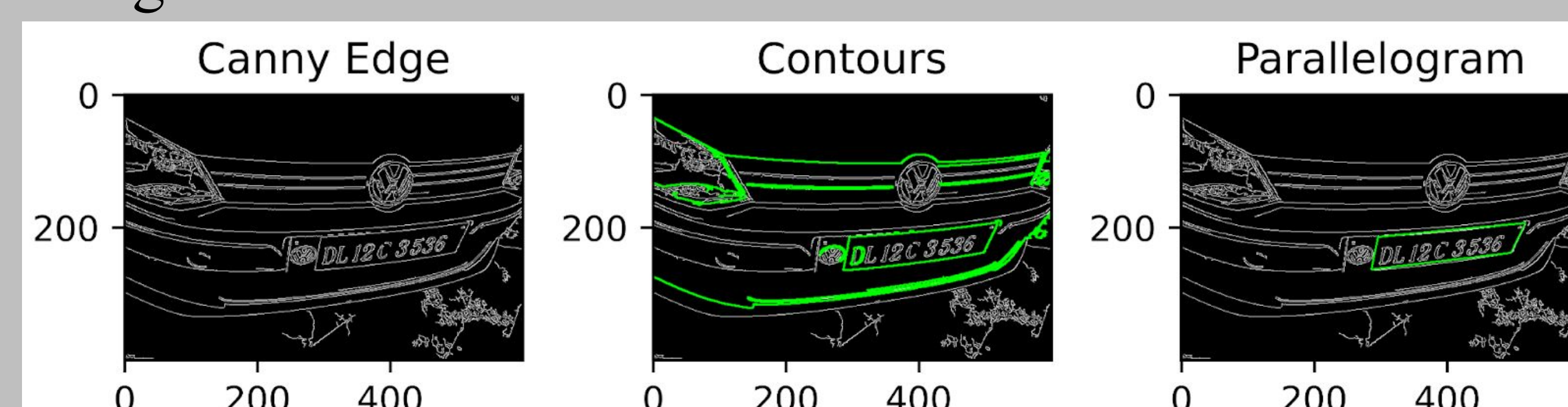
1. Image Pre-Processing

We will convert the image into grayscale and resize the image into a 600 x 400. Then we will apply bilateral filtering to remove noise.



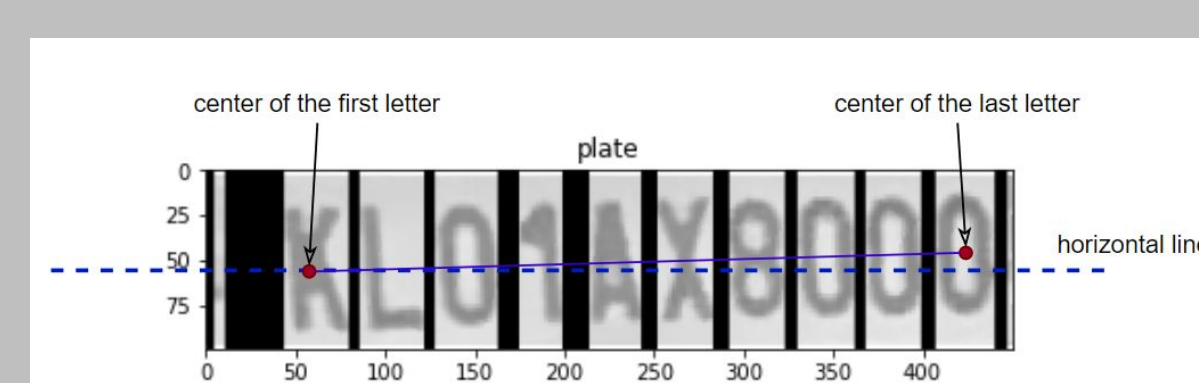
2. Car license localization

- Use Canny Edge Detection to find edges in the image
- Use cv2.findContours function to detect any possible contours in the image
- Since a car license is a parallelogram, we will use cv2.approxPolyDP to find the biggest parallelogram with 4 corners.



3. Car license rectification

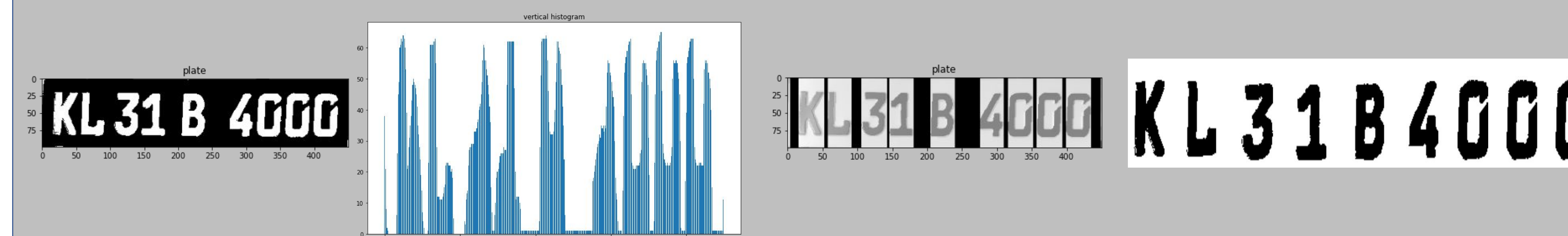
- Sort the 4 points returned by cv2.approxPolyDP
- Solve a transformation to transform the license plate area into a rectangle image
- Perform a primary letter cut, find a line that pass the centers of first and last letter
- Based on the slope of the line, perform secondary rectification



4. Letter and number template matching

4.1 Binary Image processing & letter cut

- Set up a threshold and turn the license plate into a binary image
- Draw the intensity histogram and crop all letters & numbers out
- Apply template processing to the letters & numbers that cropped out



4.2 Template matching

- Setting values of pixels of background in the letter we want to match as -1
- Setting values of pixels of background in the template we use to match as -0.2
- Setting values of all foreground pixels as 1
- Perform cross correlation between each pair of template and letter
- In cross correlation, product of two negative value is set to be 0

Performance and Accuracy

- The process of cutting letters out, which includes license plate finding, primary cut, secondary refinement and secondary letter cut, is very fast and can be computed in a few seconds.
- We can get nearly perfect letter cut if we find suitable parameters for each image.
- we need about one minute to compute the result for template matching
- For the all 97 test images, the program can find possible contours from 54 of them. And from these 54 images, the program succeeds in finding locations of car plates in 44 of them. And from these 44 images, 40 of them are successful to crop the letters out by the program. For the template matching results based these 40 images, most letters and numbers have an average success rate of 65.2%. For the letters like 'B', '6', 'G', '0' and 'D', the average success rate is much lower than other letters.

Reference

- <https://www.kaggle.com/ckay16/indian-number-plate-detection>
- <https://www.pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-an-pr-with-python/>
- <https://medium.com/programming-fever/license-plate-recognition-using-opencv-python-7611f85cdd6c>