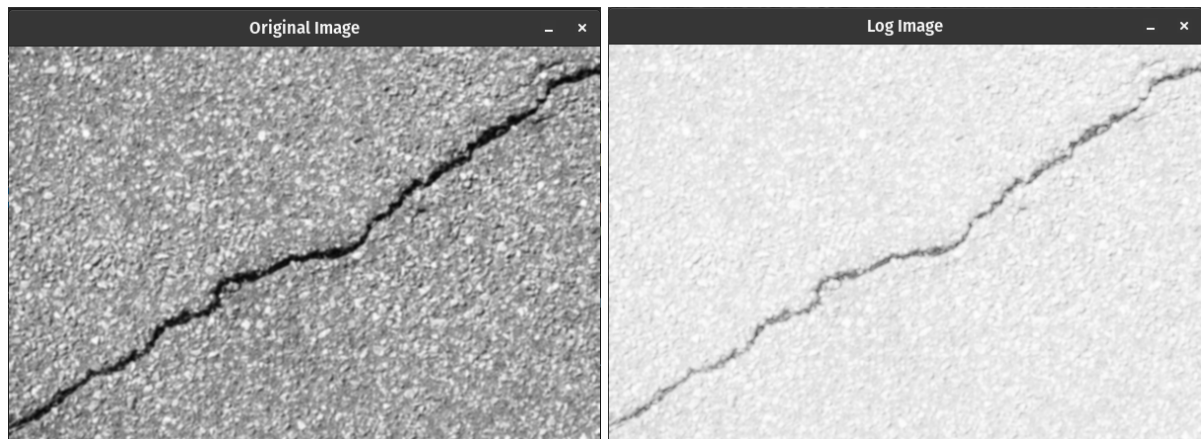


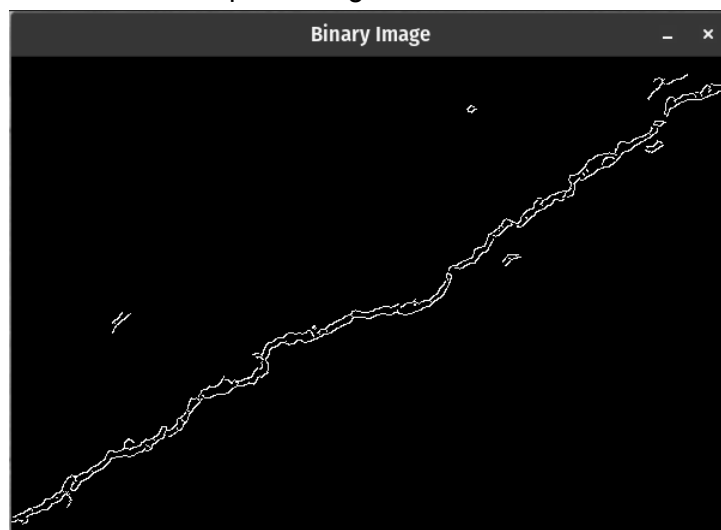
Computer Vision - Report for lab 5

Task 1

For this task I have done some trials with my colleagues, getting discrete results. We tried applying Gaussian blur filters with different sizes to the images, but nothing was working good enough for all the three images. We have also tried to normalize the image, and we have tried different segmentation techniques, such as watershed, bilateral filter and otsu filter. In the end, after some research on the internet we have found this repository <https://github.com/shomnathsomu/crack-detection-opencv>. It is a small "guide" on how to detect cracks using opencv, however it wasn't perfect since every image is unique, but I have taken inspiration from it. So, the final process is: blur the image, apply log transform, normalize the image and apply a bilateral filter. In the end, the program detects the cracks through Canny edge detection. The use of the log transformation was definitely a good idea since it helps a lot to enhance the contrast of the image, and it removed a lot of "noise" around the real crack. This is the comparison before and after:



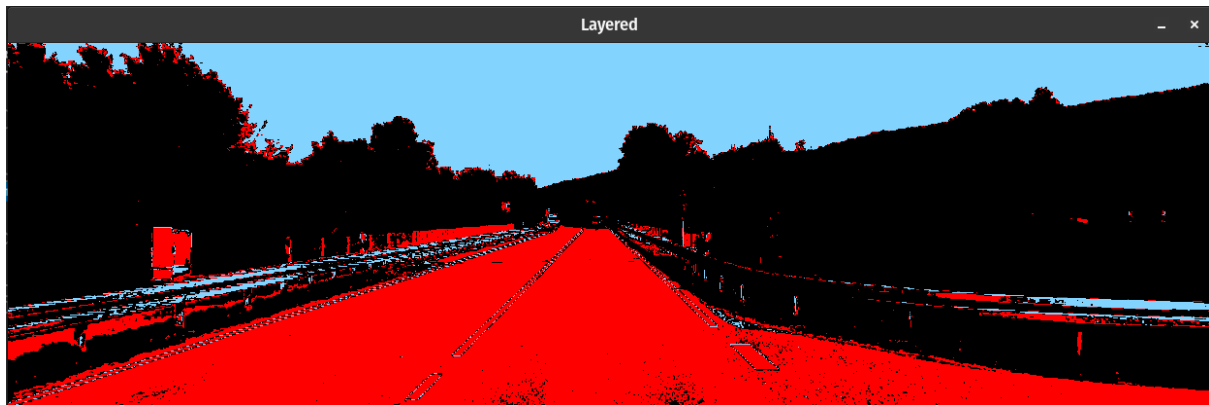
This is the final result for the first asphalt image:



Task 2

In this task, I have used the same technique I have used in task4 of lab3. I have selected an initial color based on a triplet of RGB values. Then, for each pixel of the image, I have computed the mean value of all the pixels inside the filter centered on the inspected one.

Subsequently, I selected the pixels that returned a difference between the obtained mean and the reference RGB value chosen at the beginning that is lower than a threshold value. I have adjusted the parameters and therefore I have created three masks: sky, asphalt and lines (because lines are almost white). In the end, I have merged together asphalt and lines in order to form the road mask. The red part is the road, the light-blue one is the sky and the black one is everything else:



Task 3

In this task I have used the same segmentation method used in the previous task. Adjusting the parameter, this is the final result:

