Adaptive License Plate Reading

Report on Computer Vision project 2018/2019

Roberto Rossini 1197674

June 2019

1 Introduction

This project's aim is to detect and read license plate in images. Throughout this work, I will present how I tackled this problem, which methods I used, the problems I faced and my final results with some considerations.
To run and test the code, read the `README.md` file, which explains briefly the content of the archive and how to use it. I used two programmin languages: *C++* and *Python*. Furthermore, the 2 main libraries I used are, respectively, `OpenCV` and `Keras`.
Test images (which can be found on the `cars` folder) were retrieved from the *Moodle* website and from this website.

In the first section, I will present the methods I tested to detect the license plate in the image, showing some results. I will emphasize the method I obtained best outcomes. During the second section, I will explain how I developed, trained and used a convolutional neural network to read the license plate keys.Then, I will show some results and, in the last section, I will compare the results and present some considerations.

2 License Plate Detection

In this section I will present some techniques I developed to detect a car license plate inside a given image. These techniques will be presented in increasing order of detection efficiency. However, in the final version of my code, I combined the last 2 techniques, to guarantee a larger range of detection.

2.1 Keypoints and Descriptors Object Detection

At first, I tried this method to see if it could work. I used the C++ `ObjectDection` class I had already developed to test this technique. The class simply load 2 images: an object image and a scene image. The class uses the following methods to perform the object image detection inside the scene image:

- `compute()`: this method computes and extract the keypoints and descriptors from the images using `ORB`;
- `match()`: with this method the matches between the two images are computed;
- `refine()`: previously found matches are refined, based on the ratio argument;
- `draw()`: final method to draw a bounding box if the object has been detected inside the scene. It used `cv::findHomography` to find corners of the object in the scene image and `cv::perspectiveTransform()` to draw the box.

3 License Plate Reading

4 Results

5 Conclusions

Figure 1: not detected in `dataset4/scene1.png`.