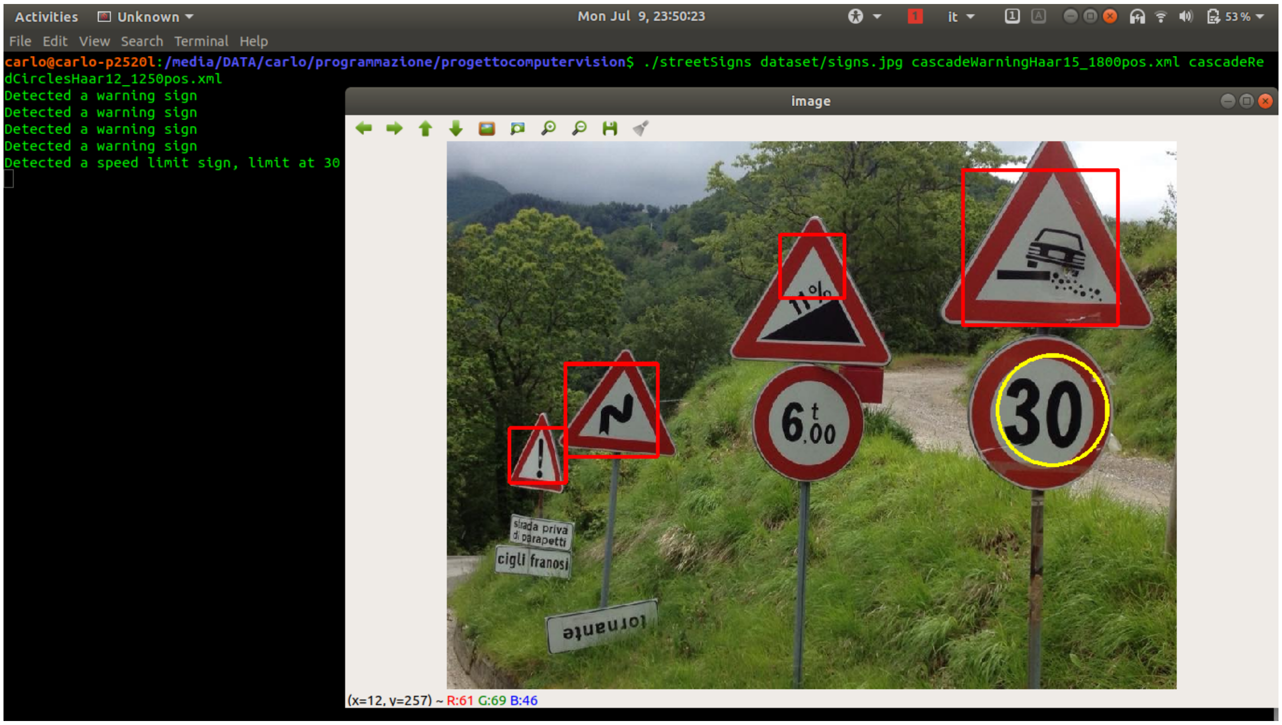
COMPUTER VISION Project 1 Autonomous Driving: Road Sign Recognition

Carlo Rizzardo - 1156404
Computer Vision

July 9, 2018

To solution to the proposed problem consists in the textitStreetSignsIdentifier class. The public interface of the class consists just in the constructor, a utility method for setting the verbosity for the whole object and the *identify(cv::Mat& img)* method. This method receives an image in input and returns a list of *StreetSign* objects which contains all the traffic signs which have been detected int the image. The *StreetSign* object is a superclass for the objects *StreetSign_Warning*, *StreetSign_NoParking* and *StreetSign_Speed*, which represent more specific types of objects. From each of these object it is possible to get the location in the image of the sign and, in the case of the speed limits, the maximum speed allowed by the sign. The actual detection and classification is performed using a variety of different methods. The first approach is the usage of two cascade classifiers, one for the triangular warning signs and the other one for all the round red signs. After this first detection the algorithm proceeds to create a mask representing the red areas in the image, this is done by two different method whose results are then intersected. The first one uses a distance function based on the hue and then multiplied by the saturation, which when thresholded provides a first mask, the other one uses a distance function based on the lab color space. Both these methods are applied to a subsampling of the image for performance reasons. The mask is then used to confirm the detections from the cascade classifiers, only the detection whose area contains at least 5% of pixels confirmed by the threshold are kept. After this the warning signs are confirmed, but the round ones need a further classification. This is done first by selecting as no-parking signs the signs which have at least a line that passes near the origin with an orientation of about 135 degrees. This detection is done using Hough on an edge image computed using the Canny edge detector. The remaining signs are then searched for digits, this is done performing first a threshold of the sign, followed by a computation of the contours and their bounding boxes these boxes are then filtered by their sizes and in them are detected the digits using a *kNearestNeighbor* classifier, which is trained with an image that can be found in the template folder.

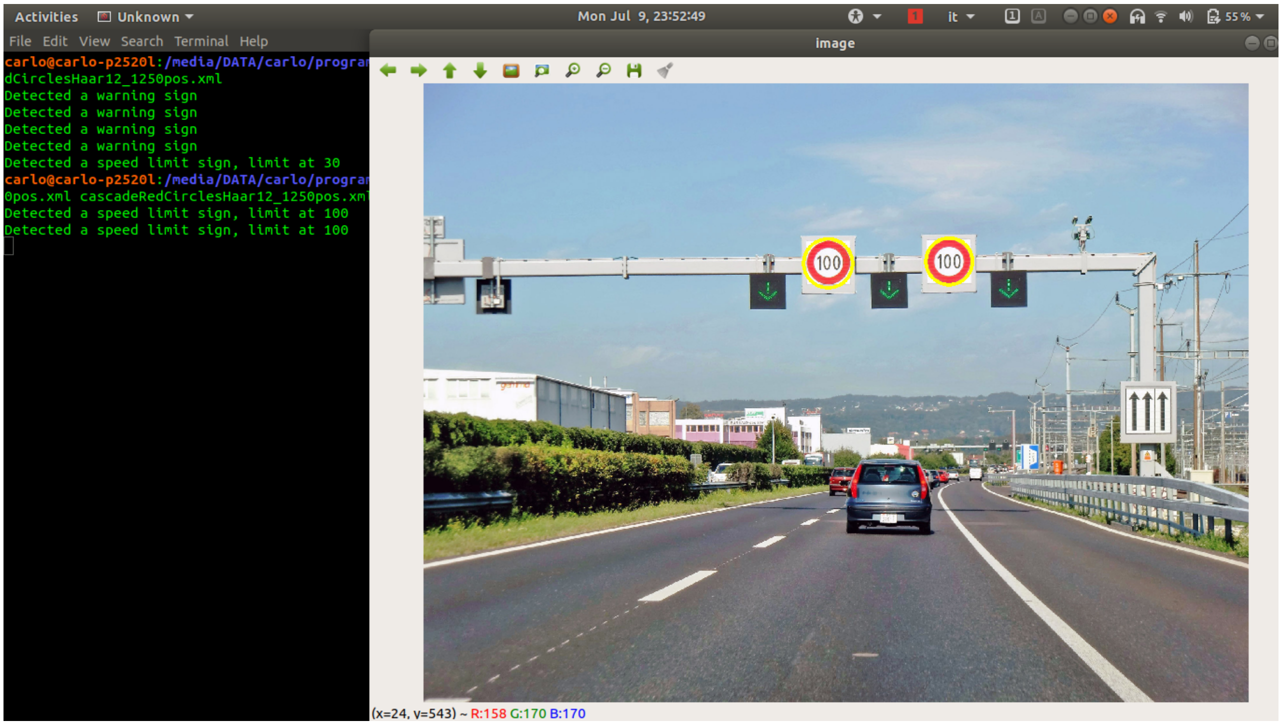
The training of the cascades has been done on a mix of samples form the BelgiumTS Belgium Traffic signs Dataset and the German Traffic sign dataset. Ultimately with 1800 positive samples for the warning signs and 1250 for the round ones.



```
carlo@carlo-p2520l:/media/DATA/carlo/programmazione/progettocomputervision$ ./streetSigns dataset/signs.jpg cascadeWarningHaar15_1800pos.xml cascadeRe
dCirclesHaar12_1250pos.xml
Detected a warning sign
Detected a warning sign
Detected a warning sign
Detected a warning sign
Detected a speed limit sign, limit at 30
```

Activities
Unknown
Mon Jul 9, 23:50:23
it
53 %
File Edit View Search Terminal Help
image
strada priva di parapetti
cigli franosi
(x=12, y=257) ~ R:61 G:69 B:46

Figure 1: The result of the algorithm on the dataset/signs.jpg image



```
carlo@carlo-p2520l:/media/DATA/carlo/progran
dCirclesHaar12_1250pos.xml
Detected a warning sign
Detected a warning sign
Detected a warning sign
Detected a warning sign
Detected a speed limit sign, limit at 30
carlo@carlo-p2520l:/media/DATA/carlo/progran
0pos.xml cascadeRedCirclesHaar12_1250pos.xml
Detected a speed limit sign, limit at 100
Detected a speed limit sign, limit at 100
```

Activities
Unknown
Mon Jul 9, 23:52:49
it
55 %
File Edit View Search Terminal Help
image
100
100
(x=24, y=543) ~ R:158 G:170 B:170

Figure 2: The result of the algorithm on the dataset/dataset/562421b3b5e3f2-18253398.jpg image