Grant agreement no: FP7-610603

European Robotic Pedestrian Assistant 2.0 (EUROPA2)

Start of the project: 01.10.2013
Duration: 3 years

DELIVERABLE 2.2

Traffic Sign Recognition

Due date: month 12 (September 2014)
Lead contractor organization: GeoAutomation
Dissemination Level: PUBLIC

1 Introduction

This deliverable describes the initial version of a traffic sign detector, developed in close collaboration between GeoAutomation and KULeuven. The final version due at M26 will report on the full performance in detection quality and speed. We implemented both a detector and a classifier that runs on a standard CPU rather than GPU, in order to avoid excessive power consumption and specialized hardware on the robot. The evaluation included both Belgium traffic signs, as detailed in the annex, and German traffic signs, in order to assess the performance of the algorithms on different data. The inclusion of different data sets already in this early stage allowed us to have a much better insight with respect to the detection issues and the influence of several parameters on the final performance.

2 Architecture

Traffic sign recognition is a very important problem for autonomous vehicles and many works are present in the literature. For a thourough overview over the existing techniques, we refer to the work of Mathias et al. [5]. The work shows that out-of-the-box methods are already able to reach very high accuracy for certain datasets, by simply employing modern variants of histograms of gradients (HOG) and sparse representations for classification [5]. However, they also observed that in order to improve the performance, those approaches are very tailored to specific data and is not clear how they generalize in other contexts.

In EUROPA2, we investigate the performance of those methods on an order of magnitude larger datasets, for both training and testing, as well as on datasets taken with adverse weather conditions, e.g., bright, dark, sun, shadow, rain, fog, night, snow. Moreover, we are interested in methodologies that relies purely on CPUs, instead of computer clusters and GPUs, for ease of scalability on generic hardware and energy consumption. As in [5], we address the problem of traffic sign recognition in two steps. First we apply a detector, learned on separately trained models, on the input image in a sliding-window fashion. In order to obtain a single detection per sign, we employ a non-maximum suppression technique to the resulting detections. Second, we classify the detected patches with a multi-class detector learned on the different types of signs.

2.1 Traffic sign detector

The traffic sign detector implemented in this work package is based on the Aggregated Channel Features (ACF) [1] framework. This framework is an extension of the Integral Channel Features (ICF) [2] method by P. Dollar, which in turn extended the famous Viola-Jones [6] architecture. Dollars ICF [2] is an extension of the Viola-Jones architecture where the features are calculated over multiple channels instead of only the gray-scale image. Typically 10 channels are used: 3 color channels, 6 gradient orientation channels and 1 gradient magnitude channel. The main extension of ACF over ICF consists in the so called aggregate image, where features are computed using a single pixel-level look-up instead of the four look-ups needed for integral images.

The use of multiple channels considerably improves the detection performance. Thus, ACF differs from ICF in the fact that features can be considered as sums of pixels in fixed-size squares instead of in variable-sized rectangles. Additionally, ACF uses fast feature pyramids for detecting objects at multiple image scales: instead of calculating the image features explicitly at each scale, they are approximated via extrapolation from nearby scales. These two extensions offer a significant speed improvement over ICF.

Figure 1: German GTSRB training set, 3 super classes

For practical reason, as also argumented in [5], the traffic signs are first organized in superclasses. The 3 main (and more obvious) superclasses, are the prohibitive, danger, and mandatory signs. They have as a common characteristics red circles, red triangles, or blue circles, respectively. For each of the superclasses, the corresponding annotated traffic signs are used as positives and trained into a detector model. During execution, the feature detection is identical for each of the traffic sign detector model that need to be verified, and therefore needs to be executed only once. In the subsequent step, each of the detectors is run separately on the feature channels.

2.2 Traffic sign classification

For the classification, we rely on the previously described detector and apply a *one-versus-all* strategy. For each possible superclass, we trained a separate classifier, where the sign under consideration is kept as a positive sample, and the rest of the collection is considered as negative. Thus, if n different subclasses need to be identified, this results in n different classifiers. The amount of subclasses might be an order of magnitude higher than the the amount of superclasses, but since the classifier is not executed on the whole image but only on the maximum responses of the previous detection stage, the additional CPU time needed for classification is negligible w.r.t. the first stages of the pipeline.

3 Experiments

Two datasets have been involved in the initial experiments. The GTSRB database from the German Traffic Sign Benchmark is publicly available. A second database is based on recorded data from GeoAutomation, consisting of a large set of Belgian traffic signs. The German set is carefully annotated and the subclasses are quite evenly represented, and is mainly focused on the 3 superclasses. For autonomous vehicle applications these can be considered as the most important ones to interpret traffic circumstances. Figure 1 gives an impression of the variability of the training set data for each of these 3 classes. The training set from GeoAutomation (fig. 2) consists of many more annotations which go beyond the typical traffic sign datasets.

In a preprocessing step, we resize the training images to a fixed square resolution. Also for annotated areas of traffic signs that are visible under perspective viewing, the distortion is compensated to fit within the same fixed square area. A border of about 5 pixels is included to allow the necessary space for feature extraction on the traffic sign borders. The detector and classifier are written in C++ and are used through a simple command line interface. The program outputs a text file with the coordinates of detection bounding of the traffic sign and its classified type.

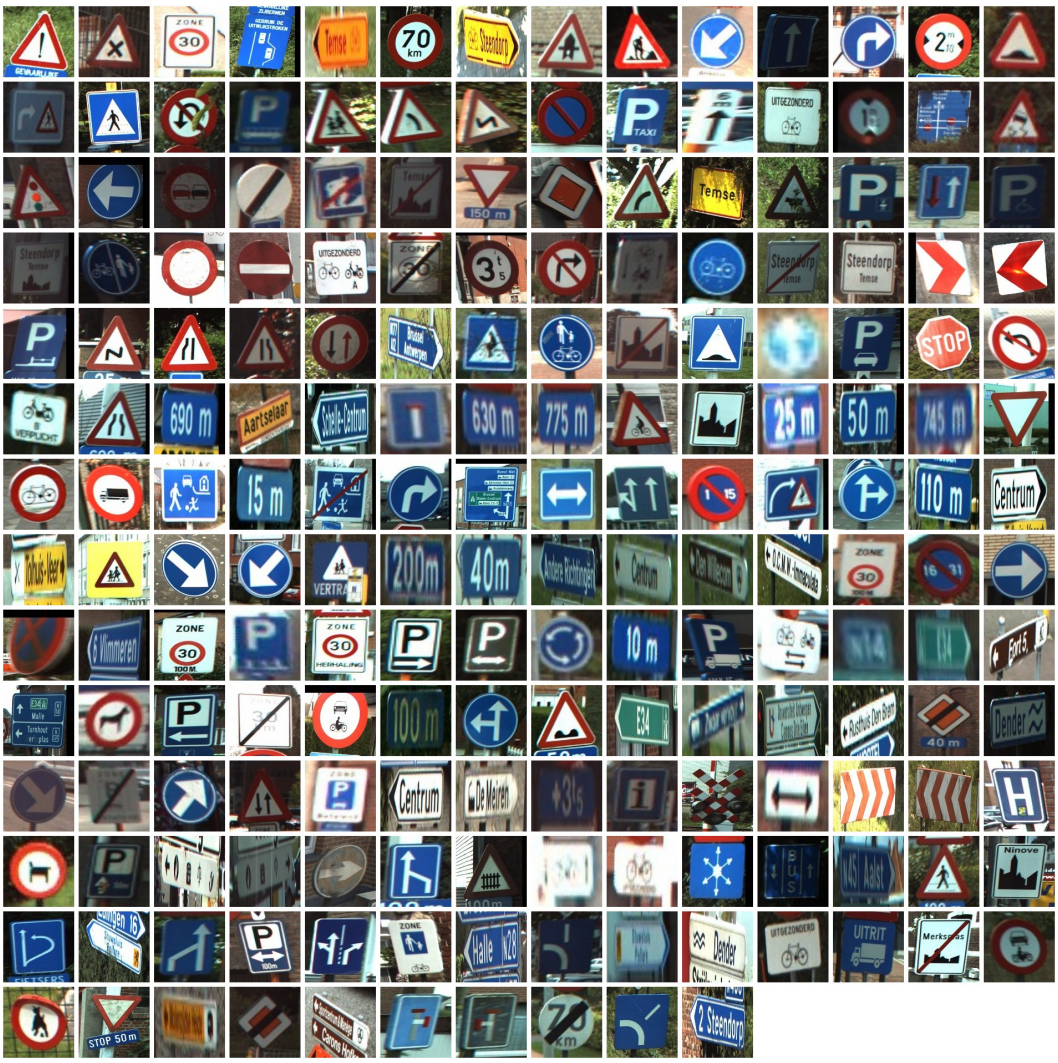

Figure 2: Belgian training set, GEOTO recording

3.1 Detection

The following parameter settings have been used in the detection pipeline.

- image resolution range from 1380x800 (German set) and 1600x1200 (Geoto)
- An image pyramid with 30 scales (ranging from 0.125 to 1.5, 8 scales per octave, of which 7 are approximated)
- A shrinking factor of 4
- Stride step 4
- 2 resolutions for each detector model, 32x32 and 64x64

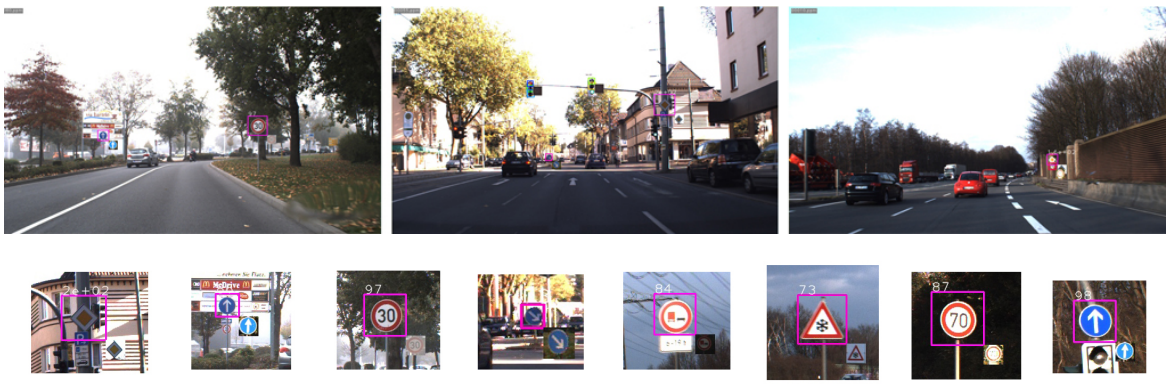

Figure 3: Detections on the German dataset

Note that for the detector model we trained for two different resolution. It turned out that the higher resolution version was not sensitive enough for traffic signs far away, and vice versa, the lower resolution model could not represent the necessary details of the traffic sign when they were visible in close-up. During the experimentation it was also observed that the performance of the classifier was depending on the accuracy of the bounding box and the scale output of the detector. Since the looping steps over positions and scale are kept to a reasonable amount for computational efficiency (stride 4) the initial detector output may show a slight misalignment of the bounding box with respect to the actual physical location of the traffic sign. This misalignment seems to affect the classification. Therefore a second detector loop is initiated nearby the location of the first detector output, this time the loops over positions and scale are refined to allow pixel resolution.

3.2 Classification

As indicated above, the classifier is defined as a set of detectors according to a one-vs-all strategy. After training, the classification performance was firs verified using a confusion matrix. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The matrix gives an indication of sensitivity of on subclass detector towards the other subclasses in the superclass. Overall, this performance was above 95%. Then the classifier models were inserted in the detection/recognition pipeline.

Figures 3 and 4 give a few examples on typical road and recording situations for the German and Belgian set respectively. The German dataset consisted of 236 frames, and overall the viewing-angle is quite orthogonal to the plane of the traffic signs. The Belgian set has 2600 frames; depending on the dataset, the viewing angle is more oblique, since the cameras are mounted on top of a van and looking downwards. The figures also show several close-ups on detections found in the sequence. The detection bounding box is color-coded with a cold-hot color map, from blue-green (lowest score) to red-magenta (highest scores). The classification is shown as a small icon at the bottom right of the bounding box, the icon image belongs to the subset that gave the maximum response during classification. During these intermediate tests, we verified the performance of the detection and classification, and came to several observations, listed below. They also justify the future work that is planned.

- The processing time for traffic signs ranged from 5 to 0.8 frames/sec. The latter which seems reasonably slower is a test on the GeAutomation training sets also involving several traffic signs

which can not be easily be put into a superclass. Although the feature extraction is typically more time demanding than the detection step, the overall detection depends on the total amount of detector models that are needed to cover the range of signs to be recognized.

- The detector is sensitive towards perspective distortions or rotations when the camera viewpoint or considerably different from the traffic sign's plane. Also occlusions remain a source of missing detections. As indicated also in [3], it can be argued that the models need to be trained with different orientations. To deal with real occlusions, more sophisticated methods are needed such as the Franken classifiers [4]. This sensitivity mainly plays a role when the sign are well visible in the image frame. On the other hand, for the smaller resolutions (far away signs), there can be an increase of false detections, such as car light or traffic lights (round shapes).
- The assumed square alignment for many traffic signs needs to be relaxed for rectangular ones.
- With respect to training stage, larger variations in contrasts are needed to handle more difficult weather conditions or brightness changes. For the German set, this was quite well accounted for, but to a lesser extent for the Belgian set.
- The current classification scheme is an extension to the ACF based detection scheme. Overall, the classifier is effective, but also seems to be sensitive to the relative amount of positive and negative annotations in the training stage (e.g. the Belgian annotations). We will also investigate alternative classification methods such as Hough Forests, Hough orchards and the diversification of features channels.

References

[1] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[2] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. In *BMVC*, volume 2, page 5, 2009.

[3] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *Computer Vision–ECCV 2014*, pages 720–735. Springer, 2014.

[4] Markus Mathias, Rodrigo Benenson, Radu Timofte, and Luc Van Gool. Handling occlusions with franken-classifiers. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1505–1512. IEEE, 2013.

[5] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition-how far are we from the solution? In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

[6] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

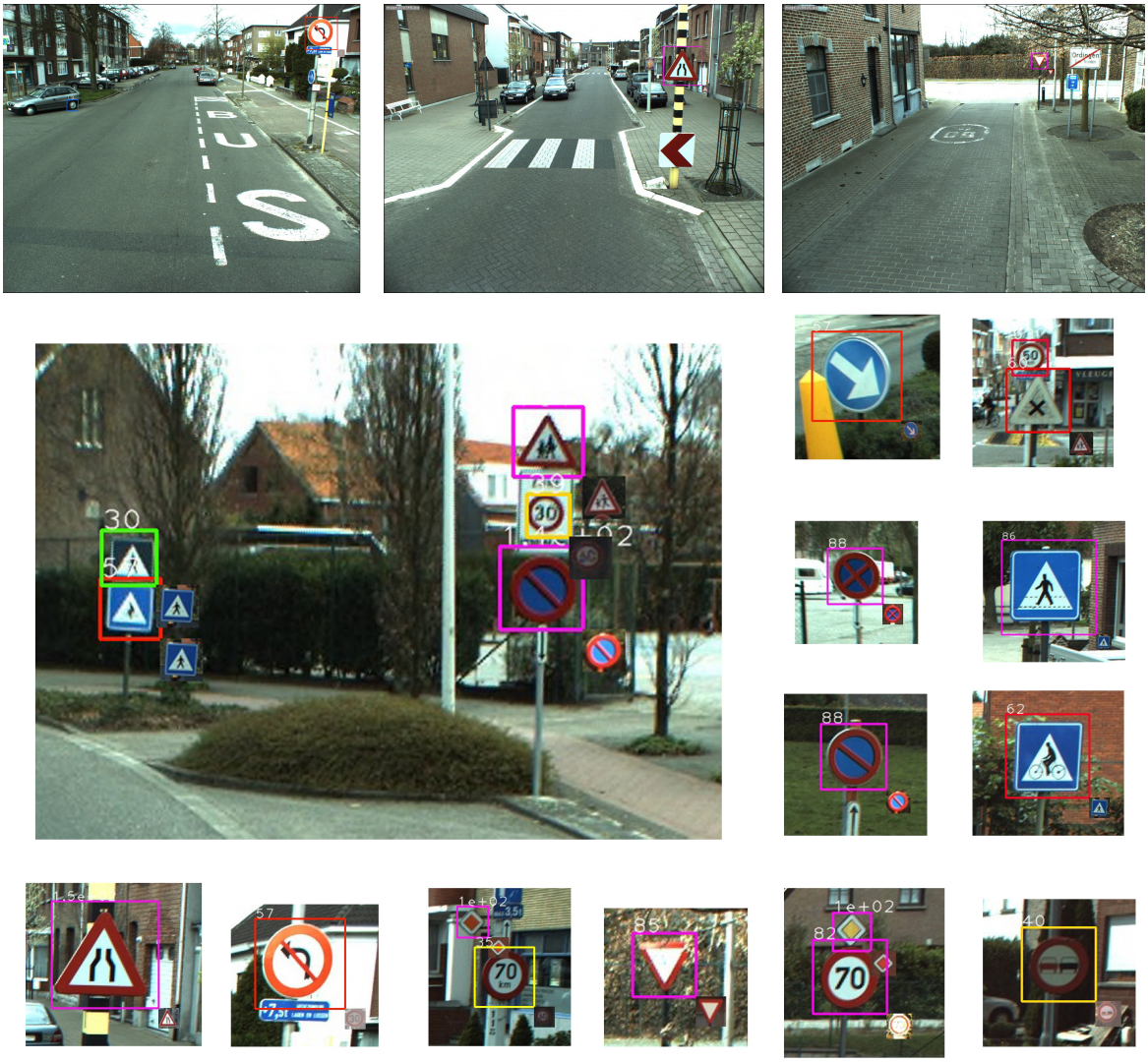

Figure 4: Detections on the Belgian dataset