Smartphone Based Mass Traffic Sign Recognition for Real-time Navigation Maps Enhancement

Bogdan Trăsnea, Gigel Măceşanu, Sorin Grigorescu, Tiberiu-Teodor Cociaş
Faculty of Electrical Engineering and Computer Science, Transilvania University of Braşov, Romania
Elektrobit Automotive Romania SRL, Braşov, Romania
Bogdan.trasnea@elektrobit.com, gigel.macesanu@unitbv.ro, s.grigorescu@unitbv.ro

***Abstract*— In this paper, a mobile device based system for recognizing speed limit and end of restriction traffic signs is proposed in the context of data acquisition for the real-time enhancing of navigation maps. Once recognized, the traffic signs are stored on a cloud server and used for updating the geospatial information of the open source OSM maps.**

The approach is mainly divided into two stages, namely, *i*) detection and *ii*) recognition. The detection is achieved through a boosting classifier, while the recognition is performed via a probabilistic Bayesian inference framework that fuses information delivered by a collection of visual probabilistic filters. The stability of the system is evaluated against a ground truth database.

Keywords— traffic sign recognition; Bayesian inference; open street maps

I. INTRODUCTION

In this paper, a speed limit and end of restriction *traffic sign* (TS) recognition system is proposed in the context of enhancing *Open Street Maps* (OSM) data used in entry navigation systems. The algorithm is targeted to run on standard commercial smartphones that can be mounted on the windshield of the car. The system detects traffic signs along with their GPS position and uploads the collected data to backend servers, via the phone's mobile data connection. The obtained TSs are further made public within the OSM community.

The recognition of road traffic signs has been a challenging problem that has engaged the computer vision community's attention for more than 30 years. According to [1], the first study of automated road sign recognition was reported in Japan in 1984. Since then, a substantial number of methods have been developed for adressing the difficulties of detecting and recognizing traffic signs. Recent increases in computing power has brought computer vision to consumer-grade applications, as stated in [2]. As computers and portable devices offer more and more processing posibilities, the goal of real-time traffic sign recognition on commercial mobile devices is becoming feasible. Traffic sign recognition is usually performed in three main steps:

- Identification of *Regions of Interest* (ROIs) containing probable TS;
- Traffic sign detection within the ROIs;
- Classification of the detected regions into traffic sign classes.

The detection of the sign is perhaps one of the most complex stages in the automatic traffic sign recongition system. In turn, traffic sign detection methods are divided into three categories: *color-based*, *shape-based* and methods based on *machine learning* techniques. For color-based detection, the obvious approach is to find ROIs based on the color properties of different areas in the input image. The main weakness here is the fact that color, as percieved by the camera sensor, is very sensitive with respect to the time of day, illumination, shadows, weather conditions etc. As an example, different color-segmentation approaches have been undertaken in [3-6]. In [5], Escalera et al. evaluated the ratios between the intensity of a given channel and the sum of all RGB channel intensities, claiming that the RGB-HSV conversion formulas are non-linear and the computational cost involved is too high. In [6], a threshold is applied over the HSV color space representation of the image with the goal of finding ROIs with a high probability of containing a TS. The most common approaches for shape-based detection of traffic signs, are the ones using different derivatives of the Hough transform, detailed in [7] and [8]. Loy and Barnes [9] have proposed a general regular polygon detector based on the so-called *fast radial symmetry*.

One of the most succesful machine learning approach for object detection in general, also intensively applied in TS recognition, has been the one proposed by Viola & Jones in [10], and followed by the discriminative version in [11]. The algorithm is based on a cascade of detectors, where each one is a chain of boosted classifiers based on Haar-like features. Also, *Support Vector Machines* [12-15] and *Deep Neural Networks* [16-18] have been used for classifying traffic signs.

The main contributions of this work can be concluded in two folds: first, this paper proposes a filtering and Bayesian inference solution for TS detection and recognition which provides optimal results on commercial mobile devices.

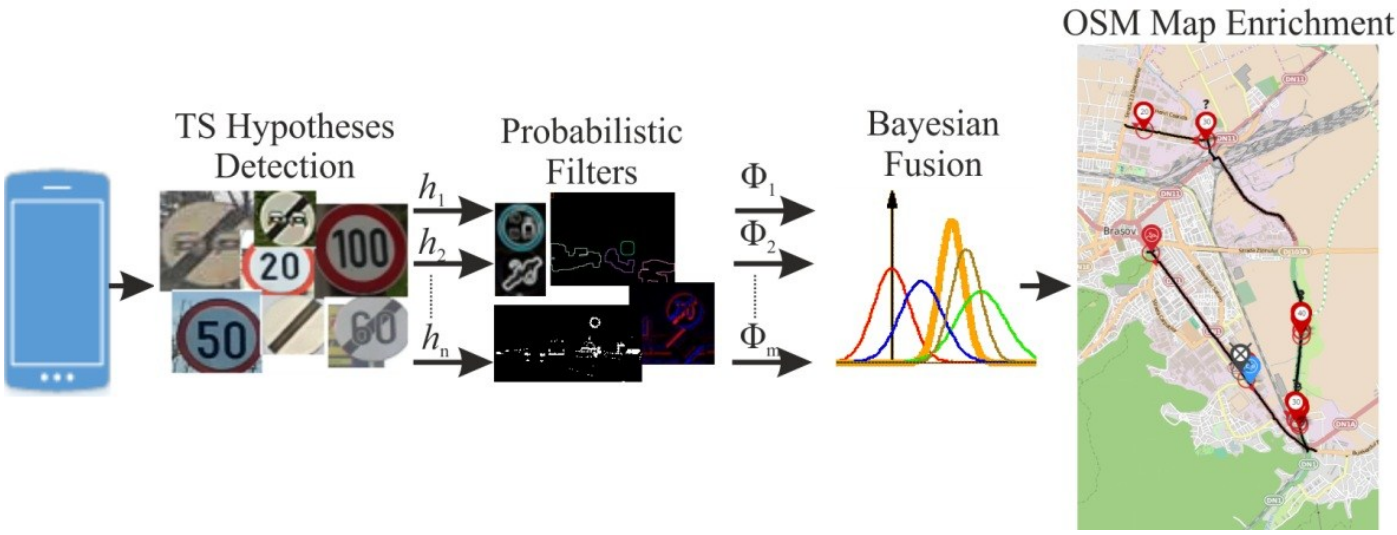

Fig. 1. Block diagram of the smartphone based traffic sign recognition system.

Second, we overcame a challenge in developing TS recognition algorithms for a broad range of mobile devices. The main challenge of this aspect is to cope with the different types of cameras included in smartphones and tablets, these cameras providing images of different quality. Although successful dedicated devices for TS recognition exist on the market, such as the MobilEye detector [19], their algorithms are developed ro run only on specific hardware.

The rest of the paper is organized as follows: in Section II, the TS recognition filters are described, followed by their fusion presented in Section III. The performance of the approach is detailed in Section IV, while conclusions are given in Section V.

II. Mobile Device Tailored Traffic Sign Recognition

The block diagram of the TSR algorithm is depicted in Fig. 1. Since the algorithm was developed to run on an Android OS, the first operation in the image processing chain is to convert the acquired image from the standard Android YUV420sp (NV21) format to its RGB counterpart. If the smartphone device is equipped with parallel processors (e.g. GPU, DSP, or multicore-CPU), then the conversion if performed using *Google Renderscript*, that is, Google's framework for the automatic parallelization of data processing. If no parallel processor is available, than the conversion if performed using the standard CPU.

A. Traffic sign detection

The obtained RGB image gets passed to the detector as a 24-bit RGB color image $I(x,y) = J^{W \times H}$ of width W and height H. (x,y) are the 2D coordinates of a pixel.

The detection process is carried out by evaluating the response of a cascade classifier calculated through a detection window sliding across the image, at different scales. The probable traffic sign ROIs are collected as the set of object hypotheses $\boldsymbol{H} = \{h_1, h_2, ..., h_n\}$. The sizes of the computed ROIs are governed by two thresholds, T_h and T_l, which constrain the sliding window to search only for signs whose sizes are within these values. T_h and T_l are chosen with respect to the probable size of a sign, as described in the *2D dimension probability filter* subsection.

From the feature extraction point of view, the classification cascade has been trained with *Extended Local Binary Patterns* (eLPB), illustrated in Fig. 2, with the local contour patterns idea found in [20]. This involves the creation of a ROI mask with a 2 pixels radius. Within the mask, which is shifted along the input TS sample, 12 neighbors are taken into consideration. A comparison of the central pixel with respect to the pixels found on the circle is performed. If the pixel on the circle has a value greater than the central one, then the value "one" is assigned to it, otherwise the value "zero". From the obtained binary result, a 12 digit number is computed. Once the procedure is applied on each pixel in the input TS image, a normalization of the obtained eLBP features is performed.

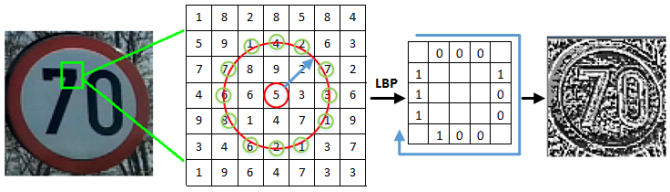

Fig. 2. Computation of the eLBP features for a 70 Km/h traffic sign sample.

B. Probabilistic filters for traffic signs recognition

The classification of the TS into speed limit classes and end of restriction signs is performed by filtering each hypothesis $h_j \in \mathbf{H}$ via a series of probabilistic filters $\phi = \{\varphi_1, \varphi_2, ..., \varphi_m\}$ and by fusing their responses within a Bayesian fusion framework, designed for TS recognition. m is the total number of filters applied. Each filter $\varphi_i(h_j)$, with $\varphi_i \in \phi$, returns a recognition probability value $p_i \in [0,1]$, $p_i \in \mathbf{P}$. Since traffic signs have different properties (e.g. speed limits over end of

restrictions), which can be better evaluated by specific filters, the response of each filter is enabled or disabled within the Bayesian fusion framework through the activation parameter $b=\{0,1\}$, associated with each $\varphi_i(h_j)$:

$$\varphi_i(h_j,b)=\begin{cases} p_i, \text{ if } b=1 \\ 0, \text{ if } b=0 \end{cases} \quad (1)$$

In the following a TS will be defined as the quantity S, where $S\in\{S_{sl},S_{eor}\}$ is either a speed limit S_{sl} or an end of restriction sign S_{eor}.

1) Position probability filter

The idea behind the position probability filter is that, using a correct smartphone setup, the traffic signs will appear in a predictable manner along a sequential series of images, as shown in Fig. 3.

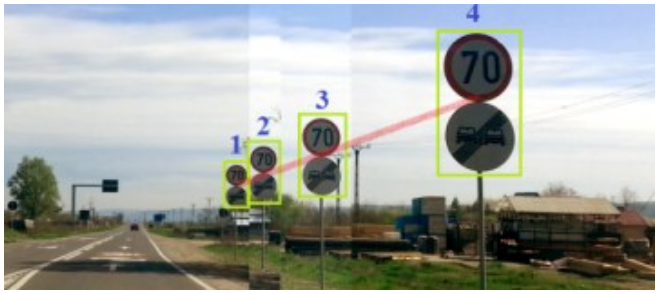

Fig. 3. Four consecutive overlayed frames which highlight the evolution of the position and dimension of a traffic sign, as the car gets closer to it.

Since traffic signs have fixed sizes, defined in each country by certain standards, we can assume that, at first, a traffic sign will appear small and closer to the center of the image. As the car will approach the sign, its dimensions will increase, while its position will be shifted to the right border of the image. In order to represent the a-priori knowledge of the probable position and size evolution of a traffic sign in a sequence of images, we have computed, from annotated training TS data, the statistical model which defines the probable evolution of a sign. The model is defined by the line equation L, as illustrated in Fig. 4.

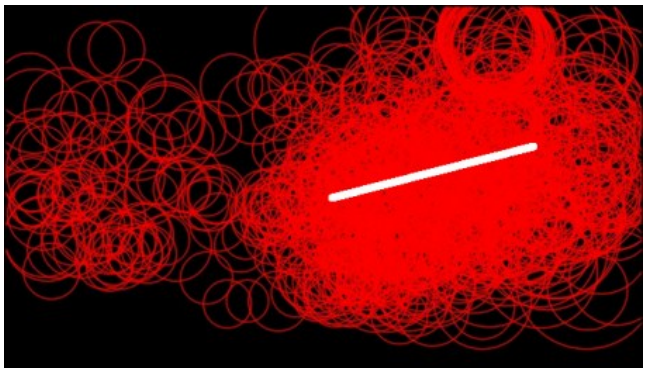

Fig. 4. Statistical model of the probable TS path in a sequence of images. Each traffic sign is represented by a red circle, while the model L is defined by the thick white line.

Considering (x_{cj},y_{cj}) as the center of the current TS hypothesis $h_j\in\boldsymbol{H}$ and d_j the projection of point (x_{cj},y_{cj}) to the model L, a position probability measure $p_1(S\mid d_j)$ can be assigned to each hypothesis $h_j\in\boldsymbol{H}$, thus obtaining a confidence value describing the probability of having a TS at a certain detection ROI in the image. Using a Gaussian probability density function, the $p_1(S\mid d_j)$ measure is quantified as:

$$p_1(S|d_j)=f(d_j,\mu_p,\sigma_p)=\frac{1}{\sigma_p\sqrt{2\pi}}e^{-\frac{(d_j-\mu_p)^2}{2{\sigma_p}^2}} \quad (2)$$

where $\mu_p=0$, since the highest detection probability is given by $d_j=0$.

2) Two-dimensional size probability filter

Along with its position, the size of a TS also varies while the car moves. We have considered that the mean value is linearly dependent on the horizontal coordinate of the sign in the image. This means that, as the horizontal coordinate increases, the mean value of the dimension also increases. Since (x_{cj},y_{cj}) is the center of the current hypothesis $h_j\in\boldsymbol{H}$, we can use the statistical equation L to compute the current mean size value $\mu_{sz}(x_{cj})$ for h_j:

$$\mu_{sz}(x_{cj})=\frac{y_2-y_1}{x_2-x_1}(x_{cj}-x_1)+y_1 \quad (3)$$

where x_{cj} is the horizontal coordinate of the hypothesis center and x_1 is the leftmost horizontal coordinate of the statistical line L. As mentioned in Section II.A, the size of a TS varies within the interval $[T_l,T_h]$.

The size of h_j has been defined as the singular quantity $s_j=(h_w+h_h)/2$, where h_w and h_h are the hypothesis width and height, respectively. The $p_2(S\mid s_j,\mu_{sz})$ value is also calculated from a Gaussian probability density function:

$$p_2(S|s_j,\mu_{sz})=f(s_j,\mu_{sz}(x_{cj}),\sigma_{sz})=\frac{1}{\sigma_{sz}\sqrt{2\pi}}e^{-\frac{(s_j-\mu_{sz}(x_{cj}))^2}{2{\sigma_{sz}}^2}} \quad (4)$$

3) Traffic sign histogram distribution filter

This filter has been designed for validating and recognizing end of restriction signs (e.g. end of speed limit, end of all restrictions, etc.), which, as with speed limits, varies in shape depending on the country of origin. The first preprocessing step is to consider the gray level image of each hypothesis $h_j\in\mathbf{H}$ (Fig. 5a). The second one is to crop the width and

height of h_j by 20%, in order to remove the noise around the rim of the sign.

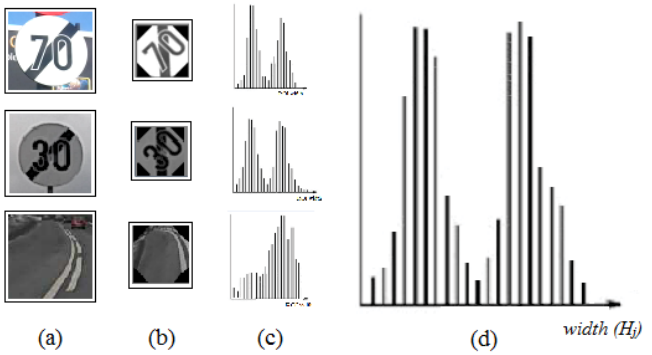

Fig. 5. Elements of the TS histogram based filter. (a) Two real and one false end of restriction hypotheses $h_j, h_{j+1}, h_{j+2} \in \boldsymbol{H}$. (b) Transformed hypotheses. (c) Histograms of h_j, h_{j+1}, h_{j+2}. (d) Reference histogram for the end of restriction sign.

The next step is to rotate the cropped image using an affine transformation in such a way that the black line will be vertical in the transformed image (Fig. 5b). Once the hypothesis has been rotated, its histogram g_j is computed along the horizontal axis, where each bin represents the sum of the pixel intensities over each column of the gray image (Fig. 5c). In order to remove outliers, g_j is smoothed with a factor f_h. As in the case of the other filters, we have created the statistical model histogram in Fig. 5d used as reference histrogram. Since the shape of the sign is defined by a centered black line having white areas on its left and right sides, the corresponding histogram pattern will have two peaks. The histogram confidence value is obtained by correlating the input TS hypothesis with the reference one. As an example, the histogram from Fig. 5c, top row, matched with the reference one in Fig. 5d, gave a match probability $p_3(S_{eor}|hist) = 0.96$. In the same time, the histogram from Fig. 5c, middle row, gave a match probability of 0.93, whereas the one from Fig. 5c, bottom row, returned a match probability of 0.27.

4) Shape based probability filter

This is also a filter that is applied for validating and recognizing end of restriction signs. The principle of detecting the skewed black line follows the idea from [21], where the goal is to detect the skewed black line l within an end of restriction sign.

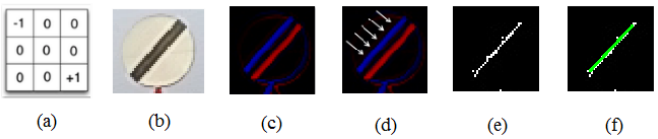

Fig. 6. Detection of the skewed black line in an end of restriction sign. (a) Diagonal derivative filter. (b) Original hypothesis. (c) Derivative image: the negative and positive gray level variations are represented by the blue and red lines, respectively. (d) Direction of scanning. (e) Pixels that follow the scanned pattern . (f) Detected line.

In [21], the entire gray level input image is filtered with the horizontal derivative filter $(-1 \quad 0 \quad 1)$. Since the skewed black line is too tilted in the image, we have found that it is more efficient to apply the diagonal derivative filter, depicted in Fig. 6a, on the input TS hypothesis for pointing out the gray level intensity variations shown in Fig. 6c. The next step is to scan the derivative image in Fig. 6c and identify the line pattern. After extracting the pixels that follow the target line model l, the *Probabilistic Hough Line Transform* (PHLT) is used to detect the line of the end of restriction sign, as illustrated in Fig. 6f. The main issues with the PHLT, related to its parameters sensitivity (e.g. distance and angle resolution of the accumulator, accumulator threshold parameter, minimum line distance and the maximum gap between the two lines), have been overcome by applying it within the $h_j \in \mathbf{H}$ ROIs. The circle c bounding the model line l is calculated using the counterpart of PHLT, that is, the *Probabilistic Hough Circle Transform* (PHCT).

For a single hypothesis $h_j \in \mathbf{H}$, containing a detected line l_j and a circle c_j, the Euclidean distance $d(c_j, l_j)$ from the center of the circle c_j to the center of the line l_j, relative to the size of c_j, has been computed. We can now assign to each TS hypothesis $h_j \in \mathbf{H}$ an occurence probability $p_4(S_{eor}|c_j, l_j)$ by using a one-dimensional Gaussian probability density function:

$$p_4(S_{eor}|c_j, l_j) = f(d(c_j, l_j), \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d(c_j,l_j)-\mu)^2}{2\sigma^2}} \quad (5)$$

where $\mu = 0$. Two examples of detected end of restrictions signs are shown in Fig. 7.

Fig. 7. Recognition of end of restriction signs in different backgrounds.

5) Extended LBP (eLBP) probability filter

The eLBP filter uses the computed local binary patterns, in combination with a *Multiclass Support Vector Machines* (Multiclass SVM) classifier, to assign a recognition probability for each TS class. The classifier maintains a model $f: \mathrm{R}^d \rightarrow \mathrm{R}^K$, which is a mapping from the input eLBP features space to the multiclass domain. The feature vector used for training the classifier is composed of the extracted eLBP features described in Section II.A. For each TS class,

the eLBP probability filter provides a confidence measure $p_5(S_{sl}|eLBP)$.

6) Color based probability filter

This filter is designed for validating speed limits traffic signs, where a color segmentation approach has been designed for detecting the red rim of this type traffic signs. The Ohta space thresholding method [5] has been used to segment the red color within the original hypothesis $h_j \in \boldsymbol{H}$. The TS colors can be classified by using the set of thresholds proposed in [5]. Once all the red pixels inside h_j have been segmented, the algorithm clusters them together in order to remove noise. The PHCT is further used for detecting circles at different radii. The output of this filter $p_6(S_{sl}|color)$ is equal to the PHCT matching probability.

III. A Bayesian Framework for Fusing TS Detection Probabilities

Once all the filters have been computed, the vector containing the returned values for each filter is passed to the Bayesian fusion block in order to obtain a final probability for each TS class, that is, speed limits and end of restrictions. The data fusion problem can be formulated as an estimation of a final traffic sign probability, calculated from all the obtained probability distributions. As described in the previous section, the involved probabilities are the following: $p_1(S|d_j)$, $p_2(S|s_j,\mu_{sz})$, $p_3(S_{eor}|hist)$, $p_4(S_{eor}|c_j,l_j)$, $p_5(S_{sl}|eLBP)$ and $p_6(S_{sl}|color)$, while $S \in \{S_{sl}, S_{eor}\}$, where S_{sl} is a speed limit and S_{eor} an end of restriction TS.

As mentioned in Section II.B, the speed limits S_{sl} and end of restriction traffic signs S_{eor} are recognized by combining the right set of filters using Eq. (1). Thus $P(S_{sl} \mid p_k)$, where k = 1, 2, 5, 6, is the final probability of having a speed limit sign when the p_k probabilities are computed, whereas $P(S_{eor} \mid p_k)$ is an end of restriction TS for k = 1, 2, 3, 4. An example for computing the fused a-posteriori estimation of the speed limits probability distribution is presented in Fig. 8.

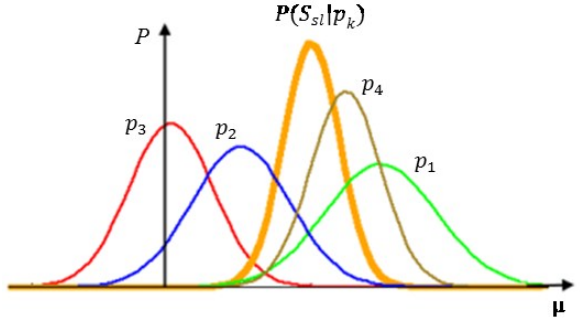

Fig. 8. The fused a-posteriori probability $P(S_{sl} \mid p_k)$ for speed limits TS.

The following fused probabilities are further obtained using Bayes rule for inference:

$$P(S_{sl} \mid p_k) = \frac{P(S_{sl})P(p_k \mid S_{sl})}{P(p_k)} \tag{6}$$

where $k = 1, 2, 5, 6$ and

$$P(S_{eor} \mid p_k) = \frac{P(S_{eor})P(p_k \mid S_{eor})}{P(p_k)} \tag{7}$$

where $k = 1, 2, 3, 4$.

Taking into account the normalization process, the denominator in (6) can be eliminated by rewritting the equation based on the conditional probability and the assumption of conditional independence:

$$P(S_{sl}|p_1, p_2, p_5, p_6) = \frac{P(S_{sl})P(p_1 \mid S_{sl})P(p_2 \mid S_{sl})P(p_5 \mid S_{eor})P(p_6 \mid S_{eor})}{P(p_k)} \tag{8}$$

$$P(S_{eor}|p_1, p_2, p_3, p_4) = \frac{P(S_{eor})P(p_1 \mid S_{eor})P(p_2 \mid S_{eor})P(p_3 \mid S_{sl})P(p_4 \mid S_{sl})}{P(p_k)} \tag{9}$$

Considering for instance the fact that the probability sum of having a speed limit traffic sign and not having one is equal to unity:

$$P(S_{sl}|p_k) + P(\overline{S_{sl}}|p_k) = 1, \tag{10}$$

by applying Bayes' rule on the left side of (10) and eliminating the denominator, we will end up with:

$$P(p_k) = P(S_{sl})P(p_k^{h_j}|S_{sl}) + P(\overline{S_{sl}})P(p_k \mid \overline{S_{sl}}), \tag{11}$$

while for the end of restriction traffic sign we have:

$$P(p_k) = P(S_{eor})P(p_k|S_{eor}) + P(\overline{S_{eor}})P(p_k \mid \overline{S_{eor}}) \tag{12}$$

The final estimate for a TS is thus calculated by replacing equations (11) and (12) in (8) and (9).

IV. Experimental Results

The described experiments are focused on the evaluation of the detection and recognition stages which make up the TS recognition system, as well as on the overall system tests. The algorithm has been tested in two countries, that is, Germany and Romania.

Along with the Android App drive tests, offline tests have been performed on a database of 4000 images collected with various types of commercial mobile devices (e.g. smartphone and tablets). All images have been manually annotated using a tool created for this purpose.

The following information, summarized in Table I, has been calculated within the testing procedure:

- *Number of false positive* (FP) – detected hypotheses containing no real traffic signs;
- *Annotated but not detected* (ND) – traffic signs that are not detected in the testing dataset, but are annotated;
- *Number detected traffic signs* (DS) – total number of detected traffic signs, also called true positives;
- *Recognition rate* (RR) – percentage estimate of correctly detected traffic signs.

TABLE I.

Test / Country	FP [nr. of traffic signs]	ND [nr. of traffic signs]	DS [nr. of traffic signs]	RR [%]
Romania	81	52	1734	85.98
Germany	74	97	1682	80.23

The recognition rate RR is calculated from a so called *confusion matrix* which encodes the true and false positives of each TS class into a 2D matrix having the same number of lines an columns. Both the lines, as well as the colums, correspond to the number of TS classes. As an example of confusion matrix computation, the index [2, 1] is incremented if a traffic sign of class 1 (e.g. 5 km/h) is classified by the algorithm as belonging to class 2 (e.g. 10 km/h). Ideally, the confusion matrix should have positive values only on its main diagonal, meaning that all the detections are precise. In order to cope with annotated, but undetected TSs, we have added to the confusion matrix an extra line and column which's elements get incremented whenever an annotated sign is not recognized. A couple of recognized TS, together with their ROIs and Bayesian inferred recognition probability, are presented in Fig. 9.

Fig. 9. Examples of recognized TS together with their Bayesian inferred recognition probability

V. Conclusions

In this paper, a traffic sign recognition system designed for commercial mobile devices is proposed. Its main goal is to acquire mass TS data which can be latter used for enhancing OSM based navigation maps.

As opposed to classical hardware oriented TS recognition systems, the presented approach has been designed to optimaly function on a broad scale of mobile devices. As future work, the authors consider the further extension of the algorithm to other types of TS (e.g. city limits, dynamic TS, give way, etc.), as well as their processing on the backend server on which the mass TS data is stored. The objective of the backend processing is to obtain a 100% accurate TS information within the navigation database. This performance can be derived from the high number of samples which belong to a single real TS instance.

References

[1] H.G. Moreno, S.M.. Bascon, P.G. Jimenez, and S.L. Arroyo, "Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition," IEEE Transactions on Intelligent Transportation System, 2010, vol 11, No 4, DOI: 10.1109/TITS.2010.2054084.

[2] M. Mathias, R. Timofte, R. Benenson, L. Van Gool, Traffic sign recognition how far are we from the solution? in: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, Dallas, TX, USA, 2013, pp. 1–8, DOI: 10.1109/IJCNN.2013.6707049.

[3] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of CVPR, IEEE, Columbus, OH, USA, 2014, pp. 580–587, DOI: 10.1109/CVPR.2014.81.

[4] H. Li, F. Sun, L. Liu, L. Wang, "A novel traffic sign detection method via color segmentation and robust shape matching", Neurocomputing 169 (2015) 77–88.

[5] A. De la Escalera, J. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," Image Vis. Comput., vol. 21, no. 3, pp. 247–258, Mar. 2003.

[6] I.M. Creusen, R.G. Wijnhoven, E. Herbschleb, P. De With, Color exploitation in hog-based traffic sign detection, in: Proceedings of ICIP, IEEE, Hong Kong, China, 2010, pp. 2669–2672, DOI: 10.1109/-ICIP.2010.5651637.

[7] R. Timofte, K. Zimmermann, L. Van Gool, Multi-view traffic sign detection, recognition, and 3D localisation, Mach. Vis. Appl. 25 (3) (2014) 633–647.

[8] F. Ren, J. Huang, R. Jiang, and R. Klette, "General traffic sign recognition by feature matching," in Proc. 24th Int. Conf. IVCNZ, Nov. 2009, pp. 409–414.

[9] G. Loy, "Fast shape-based road sign detection for a driver assistance system," in In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, 2004, pp. 70–75), DOI: 10.1109/IROS.2004.-1389331.

[10] P. Viola and M. Jones, "Robust real-time object detection," in International Journal of Computer Vision, 2001.

[11] Chen, S. Lu, Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression, IEEE Trans. Veh. Technol. 65 (6) (2016) 4006–4015, DOI: 10.1109/TVT.2015.2500275.

[12] F. Zaklouta, B. Stanciulescu, Real-time traffic sign recognition in three stages, Robot. Auton. Syst. 62 (1) (2014) 16–24.

[13] F. Boi and L. Gagliardini, "A support vector machines network for traffic sign recognition," in Proc. IJCNN, 2011, pp. 2210–2216.

[14] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-sign detection and recognition based on support vector machines," Intelligent Transportation Systems,

IEEE Transactions on, vol. 8, no. 2, pp. 264–278, June 2007, DOI: 10.1109/TITS.2007.895311.

[15] J. Lillo-Castellano, I. Mora-Jimenez, C. Figuera-Pozuelo, J. Rojo-Alvarez, Traffic sign segmentation and classification using statistical learning methods, Neurocomputing 153 (2015) 286–299.

[16] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In CVPR, 2012.

[17] Y. Zhu, C. Zhang, D. Zhou, "Traffic sign detection and recognition using fully convolutional network guided proposals", Neurocomputing 214 (2016) 758–766.

[18] P. Sermanet, Y. LeCun, Traffic sign recognition with multi-scale convolutional networks, in: The 2011 International Joint Conference on Neural Networks (IJCNN), IEEE, San Jose, California, USA, 2011, pp. 2809–2813, DOI: 10.1109/IJCNN.2011.6033589.

[19] STMicroelectronics & Mobileye Deliver 2nd-Generation SOC for Vision-based Driver Assistance, NewswireToday Amstelveen, Netherlands, May 20, 2008.

[20] Parada-Loira, F.. "Local contour patterns for fast traffic sign detection", in: Proceedings of the Intelligent Vehicles Symposium (IV) IEEE, University of Vigo, Vigo, Spain Alba-Castro, J.L. 2010, DOI: 10.1109/IVS.2010.5548008.

[21] C. Caraffi, E. Cardarelli, P. Medici, P. P. Porta, G. Ghisio, and G. Monchiero. "An algorithm for italian derestriction signs detection," in Proceedings of IEEE Intelligent Vehicles Symposium, pages 834–840, 2005, DOI: 10.1109/IVS.2008.462130.