Robotics and Autonomous Systems 62 (2014) 16–24

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Real-time traffic sign recognition in three stages

Fatin Zaklouta *, Bogdan Stanciulescu

Robotics Center, Mines ParisTech, 60 Bd Saint-Michel, 75006 Paris, France

ARTICLE INFO

Article history:
Available online 8 August 2012

Keywords:
Traffic Sign Recognition (TSR)
Advanced Driver Assistance Systems (ADAS)
Intelligent transport systems
Color segmentation
Feature space reduction
German Traffic Sign Recognition Benchmark (GTSRB)

ABSTRACT

Traffic Sign Recognition (TSR) is an important component of Advanced Driver Assistance Systems (ADAS). The traffic signs enhance traffic safety by informing the driver of speed limits or possible dangers such as icy roads, imminent road works or pedestrian crossings. We present a three-stage real-time Traffic Sign Recognition system in this paper, consisting of a segmentation, a detection and a classification phase. We combine the color enhancement with an adaptive threshold to extract red regions in the image. The detection is performed using an efficient linear Support Vector Machine (SVM) with Histogram of Oriented Gradients (HOG) features. The tree classifiers, K-d tree and Random Forest, identify the content of the traffic signs found. A spatial weighting approach is proposed to improve the performance of the K-d tree. The Random Forest and Fisher's Criterion are used to reduce the feature space and accelerate the classification. We show that only a subset of about one third of the features is sufficient to attain a high classification accuracy on the German Traffic Sign Recognition Benchmark (GTSRB).

1. Introduction

Advanced Driver Assistance Systems (ADAS) play an important role in enhancing car safety and driving comfort. Some of their components include navigation systems to provide directions as well as up-to-date traffic information and vision-based systems such as lane departure warning systems and Traffic Sign Recognition (TSR).

The latter enhances traffic safety by informing the driver of speed limits or possible dangers such as icy roads, imminent road works or pedestrian crossings. TSR algorithms face three main difficulties:

i. poor image quality due to low resolution, bad weather conditions, over- or under-illumination,
ii. rotation, occlusion and deterioration of the signs,
iii. limited memory and processing capacities in real-time applications such as ADAS.

Some examples of the first two difficulties are illustrated in Fig. 1.

We start this paper by giving an overview of the existing TSR approaches in Section 2. Our three stage approach is described in Section 3 and the overall performance of the proposed system is presented in Section 4. The effect of the feature selection on the performance of the traffic sign classification is evaluated in Section 5. We conclude this paper and present an outlook on further possible improvements in Section 6.

2. Existing approaches

Most traffic recognition algorithms divide the problem into three stages:

i. a rough segmentation to determine the location of the signs
ii. category determination and
iii. candidate classification to identify the content of the extracted traffic signs using various machine learning techniques.

The segmentation phase is not a mandatory step. However, it is often deployed in approaches using color images. This section gives a brief overview of some of the techniques used in the three TSR stages.

Some approaches, such as [1,2] reduce the memory and processing requirements by using tracking. The candidates found are tracked over several frames to reinforce the decision made or to reduce the search space in the subsequent frame and therewith accelerate the performance. Some implementations, such as [3], only examine every n-th frame, to speed up the overall system.

2.1. Determining the location using segmentation

As traffic signs need to be easily perceivable, they are brightly colored in red, blue, yellow. Hence, the detection is often based on a pre-segmentation of the image to reduce the search space and retrieve Regions of Interest (ROIs).

Since the direct thresholding of the RGB channels is sensitive to changes in illumination, the relation between the RGB (Red Green Blue) colors is often used. In [3], the color enhancement is used to extract red, blue and yellow blobs. This transform emphasizes the

* Corresponding author.
E-mail addresses: fatin.zaklouta@ensmp.fr, fatin.zaklouta@mines-paristech.fr (F. Zaklouta), bogdan.stanciulescu@mines-paristech.fr (B. Stanciulescu).

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.07.019

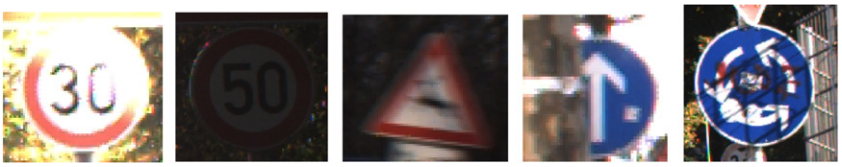


Fig. 1. Examples for difficulties facing the Traffic Sign Recognition (TSR) task: over-illumination, under-illumination, rotation, occlusion and deterioration of the signs.

pixels where the given color channel is dominant over the other two in the RGB color space.

In [4], chromatic and achromatic filters are used to extract the red rims and the white interior of the speed limit and warning traffic signs respectively. The HSI model (Hue Saturation Intensity) is used in [5] as it is invariant to illumination changes. Empirically determined fixed thresholds define the range of each HSI channel in which lie the red and blue traffic sign candidates. It is pointed out in [6], however, that HSI is computationally expensive due to its nonlinear formulae.

2.2. Category determination using shape detection

Several detection algorithms are based on edge detection, making them more robust to changes in illumination. These are shape-based methods which exploit the invariance and symmetry of the traffic signs. They are however often sensitive to the quality of the preprocessing, such as edge extraction or color segmentation.

Franke et al. [7] use Distance Transform (DT) and Template Matching (TM) to detect circular and triangular signs. Similarly, Ruta et al. [3] use the Color Distance Transform, where a DT is computed for every color channel separately. The advantage of matching DTs over edge images is that the similarity measure is smoother and robust to slight rotations. It is, however, sensitive to affine rotations and occlusions.

In [4], four Support Vector Machines (SVMs) are trained on the Distance to Border (DtB) vectors to classify the shape of an extracted candidate ROI. In [1], the FFT signatures of candidate signs are used to detect relevant shapes. This feature is robust to rotation and scaling, yet not to occlusion and deterioration.

The Hough Transform is also widely used to detect regular shapes such as circles and triangles [8,9,5]. The processing time is decreased by the simpler Radial Symmetry Detector [10], yet it is limited to circular traffic signs. Ruta et al. [9] refine the Hough Transform result using the Confidence-Weighted Mean Shift to eliminate redundant detections. The Hough transform is combined with an iterative process of median filtering and dilation to refine the candidate set in [5].

Many recent approaches use gradient orientation information in the detection phase. Gao et al. [11] classify the candidate traffic signs by comparing their local edge orientations at arbitrary fixation points with those of the templates. In [12], Edge Orientation Histograms are computed over shape-specific subregions of the image. In [13,14], the Regions of Interest (ROI) obtained from color-based segmentation are classified using the Histogram of Oriented Gradients (HOG) feature. To integrate color information in the HOG descriptor, Creusen et al. [15] concatenate the HOG descriptors calculated on each of the color channels. The advantages of this feature are its scale-invariance, the local contrast normalization, the coarse spatial sampling and the fine weighted orientation binning.

2.3. Classification techniques

The classification techniques used to determine the content of the detected traffic signs belong to two general categories: learning and Nearest Neighbor approaches. The learning consists of finding an optimal separation between two or more classes. It includes, amongst others, one-vs-all SVM classifiers, Adaboost and Neural Networks. The Nearest Neighbor approaches seek the most similar existing training sample to the given unknown. They include template matching and tree classifiers such as K-d trees and Random Forests.

Xie et al. [13] train the HOG descriptors of each class using one-vs-all SVMs. The Forest-ECOC Adaboost classifiers achieved high performance rates in [16]. Multi-layer Perceptrons (MLPs) yield high accuracy rates in [17,18]. They also achieve low false positive rates when identifying the characters in speed limit signs in [19]. The performance of the Neural Networks is improved by pre-selecting the color–shape features using PCA and Fisher Linear Discriminant in [20].

The K-d tree is used in [5] to identify the content of the sign. The Random Forests used in [21] outperform the one-vs-all SVMs on their dataset. They also generate an ensemble of SVMs using bagging and a boosted ensemble of Naive Bayes classifiers, which improve the performance of the non-ensemble version. However, these do not outperform the Random Forests.

The advantage of the tree classifiers is that they are easy to train and update. The learning approaches tend to be biased towards over-represented classes and generally require a large training set. The Best Bin First Approximate Search [22] enables a rapid search in the K-d tree. The randomness and ensemble voting of the Random Forests make them robust to outliers and unbalanced datasets.

3. Our approach

We propose a speed and danger traffic sign recognition approach consisting of three stages: (i) segmentation, (ii) shape detection and (iii) classification. These are illustrated in Fig. 2. The image segmentation reduces the search space to the red regions that potentially contain a traffic sign. We improve the red color enhancement approach used by Ruta et al. in [3] by introducing a global threshold. The circular and triangular signs are detected using one linear SVM/HOG detector each. The candidates found are further efficiently classified using tree classifiers. We further introduce spatial weighting techniques to improve the accuracy and feature space reduction for resource optimization.

The performance of each of these three stages is compared to the state-of-the-art techniques. The classification step was also put to the test at the live German Traffic Sign Benchmark Challenge [23], where our approach was ranked 3rd in terms of accuracy and proved to be suitable for embedded systems as it runs in real-time and is resource efficient.

3.1. Image segmentation

Potential ROIs are extracted in the segmentation phase. We implement the red color enhancement, proposed by Ruta et al. in [3,24], the chromatic filter [4] and introduce the morphological filters. After applying one of these filters, the image is thresholded using an empirically determined threshold or the adaptive threshold which we propose. The resulting binary mask is used to

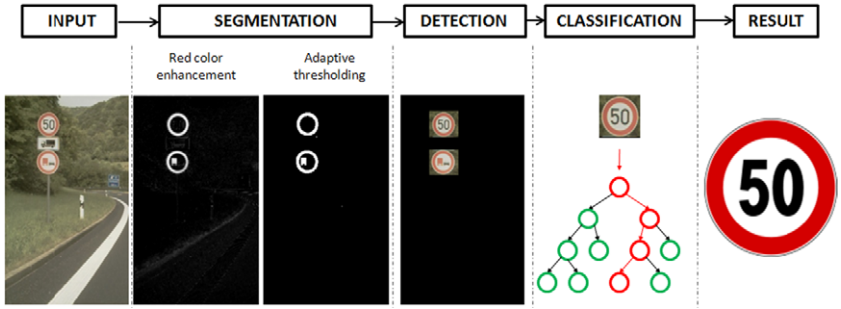


Fig. 2. Three stage approach proposed for Traffic Sign Recognition (TSR): (i) segmentation (ii) detection using HOG/SVM, (iii) classification using tree classifiers.

reduce the search space and reduce the number of false alarms of the successive detector.

3.1.1. Color enhancement

For each RGB pixel $x = \{x_R, x_G, x_B\}$ in the image, the red color enhancement is provided by

$$f_R(x) = \max\left(0, \frac{\min(x_R - x_G, x_R - x_B)}{s}\right) \quad (1)$$

$$s = x_R + x_G + x_B. \quad (2)$$

The pixels having a dominant red component are extracted, while all others are set to zero. In their approach, the ROIs are found in a subsequent stage by recursive thresholding using a Quad-tree.

In our approach, we accelerate the segmentation stage by eliminating the recursion and applying a global threshold to the enhanced image to generate a binary mask of the ROIs. The threshold used is empirically set to $\mu + 4 \cdot \sigma$, where μ and σ are the mean and standard deviation of the pixel values over the entire filtered image. The global mean helps take into account the illumination of the whole image, while a mean computed locally is more sensitive to small illumination variations in the image.

Fig. 3b shows the result of the red color channel enhancement. Note that the rims and the red truck pictogram are emphasized. The search space is reduced when applying the binary mask obtained from the red color enhancement. The false alarm rate is also lowered as will be shown in Section 3.2.4. The global thresholding approach is faster and more simple than the recursive thresholding using a Quad-tree. It is also robust to local variations in illumination.

3.1.2. Chromatic filter

We further evaluate the chromatic filter to determine the ROIs. Bascon et al. [4] use the achromatic segmentation masks to extract the white interior of the traffic signs and the chromatic masks for the red rims and yellow construction site signs. The chromatic color decomposition of an image is computed using

$$f(R, G, B) = \frac{|R - G| + |G - B| + |B - R|}{3D}, \quad (3)$$

where D is the degree of extraction of an achromatic color. It is empirically set to $D = 20$ in [4]. To extract the chromatic pixels, $f(R, G, B) > 1$ is used.

A threshold of $f(R, G, B) < 1$ extracts the achromatic pixels i.e. those lacking hue. However, the achromatic segmentation is not suitable for our application, since our sequences contain large white areas such as the sky and buildings. We therefore adapt the chromatic segmentation to obtain the red parts of the signs. The result of the chromatic filter is shown in Fig. 3c.

3.1.3. Morphological filters

We examine two morphological filters, top-hat and its reciprocal bottom-hat. The former emphasizes light pixels with a high contrast to their local environment, such as the inside of traffic signs. The latter emphasizes dark pixels with a high contrast to their local environment such as sign rims and some pictograms. The images obtained by these two operators, I_T and I_B respectively, are defined as

$$I_T = I - I_O, \quad I_O = I_D \circ I_E \quad (4)$$

$$I_B = I_C - I, \quad I_C = I_E \circ I_D \quad (5)$$

where I is the input image and I_O, I_C, I_D, I_E represent the opening, closing, dilation and erosion operators respectively.

The result of the top-hat and bottom-hat filtering is shown in Fig. 3d and e. Note that the interiors of the traffic signs are emphasized in the top-hat transform. The rims and the pictograms are emphasized in the bottom-hat transform. We further apply an empirically determined adaptive threshold $\mu + 4 \cdot \sigma$ to obtain a binary mask designating the ROIs, where μ and σ are the mean and standard deviation of the pixel values over the entire filtered image.

3.2. Category detection

In this paper, we focus on the detection of speed limit and warning signs. To evaluate the proposed detection algorithm and the performance of the color segmentation as a filter, we use image sequences acquired during the daytime in both urban and highway environments under different meteorological conditions. During the detection phase, the image is scanned at multiple scales. Only the areas resulting from the segmentation mask are examined. The resulting candidates are then passed on to the tree classifier in the classification phase.

3.2.1. Data set and evaluation

Our data set contains 24 classes of traffic signs: 12 round and 12 triangular. The training and test data sets contain 14,763 and 1584 signs respectively. There is a significant imbalance in the number of training samples for each class as the number of images per class varies between 15 and 3852. This is reflected in the real world, as some signs, such as the speed limits, are more abundant than the *wind* warning sign for example. The image resolution is 752×480 pixels.

A predicted bounding box is considered correct if it overlaps more than 50% of the ground truth. The evaluation is based on the recall and precision values, which are defined as follows:

$$\text{recall} = \frac{\text{true positives detected (TP)}}{\text{total true positives}} \times 100\% \quad (6)$$

$$\text{precision} = \frac{\text{true positives detected (TP)}}{\text{all detections}} \times 100\%. \quad (7)$$

(a) Original. (b) Enhanced. (c) Chromatic. (d) Top-hat. (e) Bottom-hat.

Fig. 3. Image segmentation for TSR: ROI extraction using red color enhancement, chromatic, top-hat and bottom-hat filters to reduce search space and accelerate detection. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2.2. Implementation of the detector

Two detectors are trained, one for the circular and another for the triangular traffic signs. The detectors are trained at an empirically determined positive:negative ratio of 1:10. The false alarms generated by the initial detector on the training dataset are reinjected as new negative samples to train a cascade classifier.

The HOG descriptor is used, as it is fast to compute and robust to changes in illumination and scale. A further reason for choosing HOG over other features is that when using undirected gradients, i.e. orientations between 0° and 180°, both static (red rim, white interior) and dynamic (illuminated, red rim, black interior) signs can be found with the same detector. Other features, such as Control Points or Haar, would fail in this case due to their use of directed gradients i.e. orientations of 0°–360°.

The computed HOG descriptors are used to train two linear SVM classifiers (one for round signs and another for triangular ones). This significantly reduces the computational costs at runtime compared to the four SVM classifiers trained per shape in [4].

The SVMLight[1] library is used. The m resulting support vectors are combined to a single global vector v. Given the SVM classification function $f(x)$ of an unknown sample x, with support vectors x_i, Lagrange multipliers α_i and labels l_i.

$$f(x) = \text{sign}\left(\sum_{i=1}^{m} l_i \alpha_i (x_i \cdot x) + b\right) \tag{8}$$

$$\Leftrightarrow f(x) = \text{sign}\left(x \cdot \sum_{i=1}^{m} l_i \alpha_i x_i + b\right) \tag{9}$$

$$\Leftrightarrow f(x) = \text{sign}\,(x \cdot v)\,. \tag{10}$$

This reduction to one support vector accelerates the detection because the HOG descriptor x of each subwindow is compared to a single vector v instead of several.

The optimal size of the HOG descriptor was determined in [25] to be a 144 value vector with a block size, stride size and cell size of 4 pixels each. It proved to be the best compromise between processing requirements and performance.

3.2.3. Use of segmentation masks

We propose to use a sliding window on the binary mask resulting from the segmentation in the detection phase. The image is resized and scanned at different scales. The sum of the pixel values $sum(img(x, y, w, h))$ in each subwindow $img(x, y, w, h)$ is computed, where (x, y) and (w, h) are the top left corner, width and height of the subwindow in the image. If the ratio of the white pixels to the area of the subwindow $area(img(x, y, w, h))$ exceeds a given threshold θ as in Eq. (11), the subwindow is considered to contain a potential sign. The integral image representation of the binary mask image is used to accelerate the sum computation.

$$\frac{sum(img(x, y, w, h))}{area(img(x, y, w, h))} > \theta. \tag{11}$$

Hence, the candidate subwindows are passed on to the HOG/linear SVM detectors for circles and triangles, which in turn verify the presence of the respective shape.

This methodology is illustrated in Fig. 4. This approach reduces the search space and accelerates the detection. As an alternative, the ROIs can be found using a connected components approach. This could, however, require a longer computation time.

3.2.4. Effect of segmentation on the detection

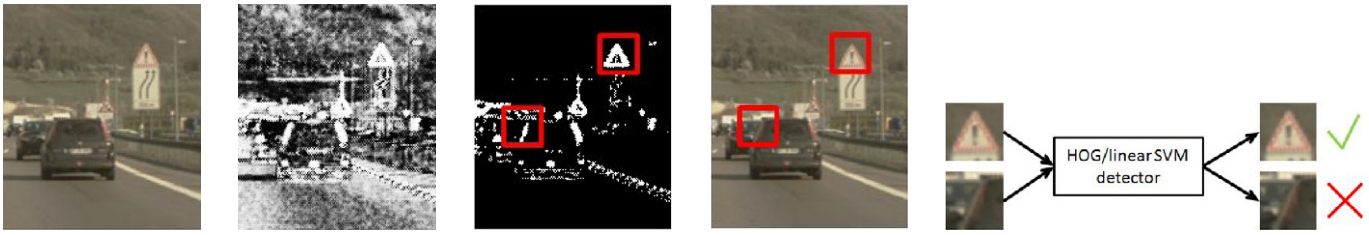
The influence of the segmentation masks on the performance of the HOG/linear SVM detectors is evaluated. The best trade-off between the recall and precision values as well as the respective average processing times per image are listed in Table 1. The red channel color enhancement improves both the recall and the precision of the HOG detector. It yields the best results with a recall of 90.21% at a precision of 90.9%. Note that the HOG descriptors computed on the RGB color space achieve a 10% higher recall than those computed on the grayscale images. This improvement is due to the strong gradients in the red color channel which are diminished in the grayscale images.

Table 1 also shows the average time (in ms) required to process a 752 × 480 pixel frame on a 2.93 GHz Intel Core i7 PC. The grayscale HOG detector achieves real-time performance, running at 35.56 ms/frame (28 Frames/s). The improved red enhancement segmentation mask requires a longer processing time, yet can run in near real-time at 48.85 ms/frame (20.5 Frames/s).

In general, the segmentation reduces the number of false positives, yet decreases the number of true detections. The morphological operators falsely eliminating signs with a low contrast to the background or within the sign due to poor illumination or deterioration of the sign. The chromatic filter omits distant and poorly illuminated traffic signs due to the lack of color richness of the corresponding pixels. The red color enhancement technique is less sensitive to changes in illumination, since it takes the relative dominance of one channel over the others into consideration.

We choose to combine the linear SVM/HOG detector with the red color enhancement segmentation technique, as it proved to be a good compromise between the resource efficiency and overall performance. The work on traffic sign recognition using segmentation masks and HOG-based SVMs presented in this section was published in [26,25].

[1] http://svmlight.joachims.org/.



(a) Input image. (b) Red color enhancement. (c) Using adaptive threshold. (d) Candidates from image. (e) Category detection using HOG/linear SVM detectors.

Fig. 4. Using the segmentation mask to find candidate ROIs: the sliding window is passed over the binary image resulting from the red color enhancement and adaptive thresholding. The extracted subwindows contain a sufficient ratio of white pixels to their area. These are passed on to the HOG/linear SVM detectors for circles and triangles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Effect of segmentation masks on HOG/lin. SVM detection performance.

Mask	HOG grayscale			HOG RGB		
	Recall	Precision	Time (ms)	Recall	Precision	Time (ms)
No mask	**75.82**	**90.16**	35.56	**84.72**	**89.89**	49.31
Top-hat	38.45	92.13	32.89	41.67	88.95	45.78
Bottom-hat	42.05	91.48	32.75	45.20	88.40	45.52
Chromatic	42.11	78.29	48.07	38.38	92.68	55.24
Red	**84.66**	**94.77**	48.85	**90.21**	**90.90**	55.54

3.3. Sign classification

The content of the candidate traffic signs detected is identified using the tree classifiers. For an extensive evaluation of the tree classifiers, we use the German Traffic Sign Recognition Benchmark [23], which contains a large number of classes and images. We replace and evaluate the Euclidean similarity measure in the K-d tree by the correlation and χ^2 distance of the HOG descriptors. Further, we introduce the spatial weighting, which is particularly adapted for the traffic sign classification. This improvement weights the feature vectors to focus the Nearest Neighbor comparison in the tree classifiers on the interior of the sign.

3.3.1. German TSR Benchmark

We further evaluate the performance of combining the HOG features with the K-d trees and Random Forests using the German Traffic Sign Recognition Benchmark [23]. The work presented in this section was published in [27].

This image data set contains 43 classes, 26,640 training images and 12,569 test images. The parameters of the three HOG features used are listed in Table 2. All the images used are resized to 40×40 pixels using a bilinear interpolation. The precalculated HOG 1–3 descriptors for the training and test images are available online.[2]

Several research teams have used the GTSRB to evaluate their algorithms. An accuracy of 99.15% was achieved using a committee of Neural Networks by Ciresan et al. [28]. They combine Multi-layer Perceptron (MLP) with deep Convolutional Neural Networks (CNN) of alternating convolutional, maximum pooling and fully connected hidden layers to ensure the robust and accurate classification of the input traffic signs. The human classifier correctly classified 98.81%. Sermanet and LeCun [29] achieved 99.17% for this same data set using Multi-Scale Convolutional Networks. They experiment with various feature subsets, network parameters and structures to optimize the results.

The memory and processing constraints in embedded systems render the large Neural Networks and extensive preprocessing and feature extraction techniques inapplicable in spite of their high recognition rates. Our approach proved to be the most suitable for embedded systems, as it runs in real-time and is resource efficient. It is described in detail and compared to other classifiers in the following.

3.3.2. Comparison of classifiers

In this section, we compare the tree classifiers, K-d tree and Random Forests, to the state-of-the-art Support Vector Machine (SVM) classifier. We also evaluate the new approaches proposed: the spatial weighting and the correlation and χ^2 as similarity measures instead of the commonly used Euclidean distance.

A. K-d tree

A K-d tree is a binary search tree organizing K-dimensional data points. Each non-leaf node splits the data into two subspaces according to the i-th feature f_i with the highest variance at that level. To ensure the tree is balanced, the median value m_i of f_i is used for the split in each node. The left subtree contains data with values $f_i < m_i$ and the right subtree $f_i > m_i$. This division is repeated until the subtrees are leaves with one sample each.

In our experiments, five nearest neighbors are retrieved from the K-d tree for each test candidate, i.e. $k_{NN} = 5$ and the number of candidates examined during the search is set to $E_{\max} = 5000$. These values are determined empirically. Unless otherwise specified the similarity measure used in the K-d tree is the Euclidean distance.

The HOG 2 descriptor has a poorer overall performance than HOG 1 and HOG 3. The former computes signed gradient orientations, i.e. 0°–360°, while the latter two use unsigned gradients, i.e. 0°–180°. When using the same number of bins, the binning is coarser in the HOG 2 descriptor i.e. the bins are larger (45° per bin) than in the HOG 1 (22.5° per bin) and HOG 3 (20° per bin) descriptors. A finer spatial binning better describes the characteristics of each traffic sign class.

[2] http://benchmark.ini.rub.de/.

Table 2
Parameters of the HOG 1–3 descriptors provided by GTSRB [23].

Name	Cell	Block	Stride	Bins	Oriented gradients	Dimension
HOG 1	5×5	10×10	5×5	8	True	1568
HOG 2	5×5	10×10	5×5	8	False	1568
HOG 3	4×4	8×8	4×4	9	True	2916

I. *Correlation as a similarity measure*

The Euclidean distance measure in the K-d tree is replaced by the correlation. The results obtained when checking five Nearest Neighbors ($k_{NN} = 5$) and examining 5000 nodes at most ($E_{\max} = 5000$) are shown in Table 3. The accuracies are improved by up to 2.44%. This improvement shows that the correlation is a more robust measure for this application, as it takes into account the trend and not the difference between the HOG descriptors.

II. *Chi Squared χ^2 as a similarity measure*

The Euclidean distance measure in the K-d tree is replaced by the χ^2 distance. The classification results obtained using this measure are shown in Table 3. There is an improvement of up to 2.84%. This shows that the χ^2 distance measure is a more suitable similarity measure for the histograms, as it relativizes the difference between the bins to their average size. This also makes it more robust to noise. The Euclidean distance measure, on the other hand, does not take into account the fact that the feature differences are not the same in all the dimensions of the feature space.

III. *Spatial weighting*

In the K-d tree the Nearest Neighbor search extracts the signs which are the most similar to the test sample. The difference between the signs usually lies in the interior region containing the pictogram or speed limit.

We propose a new spatial weighting of the features to increase the influence of the central part of the image rather than the borders. When computing the Euclidean distance between a training sample in the tree and the HOG descriptor of the test image, the difference between the individual blocks is multiplied by a factor $f \in]0, 1[$, depending on the position of the block. The interior blocks have a larger factor than the ones along the border.

The 30, 50 and 80 speed limit signs are most frequently confused by the K-d tree. The spatial weighting of the HOG descriptor values according to the location of the block improves the results of the approximate Nearest Neighbors search in the K-d tree significantly. The prioritizing of the interior helps better distinguish the fine difference between the contents of the traffic signs. As shown in Table 4, the overall classification hit rates were increased by about 15% with an $E_{\max} = 5000$.

Table 5 shows the confusion matrix of some of the largest improvements induced by the spatial weighting. The HOG 3 descriptor is used, as it achieved the best results. The classification accuracy for the 20 km/h speed limit, for example, increased by 25 samples, of which 10 were previously confused with the 70 km/h speed limit.

B. *Random Forests*

Random Forests were introduced by Breiman [30]. An extensive description is given in [31]. An ensemble of random trees forms a random forest.

To grow a random tree, a subset $I \subset I_N$ of training samples is randomly chosen with replacement. The tree is grown using this subset and is not pruned. In each node, a subset F of features is randomly chosen. The current data subset is split into I_l and I_r using the feature $f \in F$ and threshold $t \in [\min(f), \max(f)]$ with the maximum Information Gain Δ. The entropy E is calculated over the labels frequencies in I.

To classify a sample x, each of the random trees $T_j \in \{T_1, T_2, \ldots, T_T\}$ in the Random Forest is traversed. In the leaf attained in T_j, the posterior probability that x belongs to the class $l \in \{1, 2, \ldots, L\}$ is denoted by $P_j(l|x)$. The class l^* of x is determined using the combined decision of the ensemble of trees.

As shown in [27], the performance of the Random Forest does not vary significantly (0%–2%) when the parameters are changed, which makes it more generic and easier to use. The HOG 2 descriptor is used as it yields the highest accuracy of 97.2%, compared to 95.1% and 95.2% for the HOG 1 and 3 descriptors respectively. The number of trees is set to 500, the number of features and samples to 100, as these parameters yield the best results for these features.

The Random Forest achieves up to 7% higher classification rates than the K-d tree with spatially weighted blocks. Since small subsets of 100 features and 100 training samples are used to construct the random trees, the probability of choosing the HOG descriptors of the 10% border region is small and the perturbation caused by the background is less significant. Hence, the randomness of the Random Forest classifier makes it more robust to variations than the K-d tree, which uses the entire descriptor set.

C. *Multi-class SVM*

To extend our study of classifiers for TSR, the performance of the multi-class Support Vector Machines (SVM) classifiers is also evaluated. One linear SVM is trained per class to obtain 43 one-vs-all classifiers. The entire ensemble is queried and the highest confidence vote determines the class of a test sample. The Pegasos solver [32] is used, as it optimizes the training process. Table 6 shows a comparison of the results obtained on the HOG 2 feature. The multi-class SVM achieves an accuracy of 95.04%. The Random Forests outperform the SVM classifiers by 1.6%–2.1%.

Table 3
Improvement of the K-d tree classification results when using the correlation or χ^2 distance instead of default Euclidean similarity measure on the HOG descriptors. $E_{\max} = 5000$, $k_{NN} = 5$.

Weighting	HOG 1 (%)	HOG 2 (%)	HOG 3 (%)
Euclidean	74.92	73.39	75.03
Correlation	75.20	73.78	77.47
Chi squared χ^2	**76.83**	**76.01**	**77.87**

Table 4
Improvement of the K-d tree classification results when applying the spatial weighting to the HOG descriptors. $E_{\max} = 5000$, $k_{NN} = 5$.

Weighting	HOG 1 (%)	HOG 2 (%)	HOG 3 (%)
No weighting	74.92	73.39	75.03
With spatial weighting	88.53	88.73	90.39

4. Overall performance of the TSR system

In this section, the overall performance of our traffic sign recognition approach is presented. The red color segmentation and the 144 value HOG/linear SVM detector described in Section 3.2.4

Table 5
Effect of spatial weighting on classification accuracy of speed limit and warning signs with K-d tree and HOG 3 descriptor.

		True label							
		20	30	50	60	70	80	100	120
Predicted label	20	**+25**	0	0	0	0	0	0	0
	30	−5	**+132**	−29	−9	−9	−9	−3	0
	50	−6	−30	**+214**	−9	2	9	0	0
	60	0	−25	−27	**+95**	−4	−48	−5	−1
	70	−10	−11	−57	−3	**+69**	−22	−2	−2
	80	0	−41	−33	−48	−29	**+122**	−21	−26
	100	−1	−21	−35	−24	−8	−37	**+41**	−37
	120	−3	−4	−20	−1	−21	−6	−9	**+68**
	Total	60	720	750	450	660	600	480	450

Table 6
Overview of K-d tree, Random Forest and SVM classifier accuracies using the HOG 2 descriptor of GTSRB.

Classifier	Parameters	Accuracy (%)
K-d tree	$E_{\max} = 5000$, $k_{NN} = 5$	73.39
K-d tree (Correlation)	$E_{\max} = 5000$, $k_{NN} = 5$	73.78
K-d tree (χ^2)	$E_{\max} = 5000$, $k_{NN} = 5$	76.01
K-d tree (spatial weighting)	$E_{\max} = 5000$, $k_{NN} = 5$	88.73
Random Forest	100 trees, 100 var, 100 samples	96.70
Random Forest	500 trees, 100 var, 100 samples	**97.20**
1-vs-all SVM (Pegasos)	$C = 10$	95.04

were used, because they achieved the best results. The parameters of the K-d tree were set to $k_{NN} = 5$ and $E_{\max} = 1000$, as they were shown to be a good compromise of processing time, memory requirements and accuracy rate. The Random Forest used contains 100 trees. The parameters of the 1568 value HOG 1 descriptor presented in Section 3.3 were used to compute the HOG training and testing features in the classification phase.

As mentioned earlier, the combination of the red color enhancement segmentation and HOG/linear SVM detector achieves a detection rate of 90.21%. The traffic signs found are then recognized by the tree classifiers. The K-d tree with the Euclidean distance similarity measure yields a classification rate of 59.21%. The correlation and χ^2 metrics improve the accuracy slightly to achieve 61.02% and 66.13% respectively. However, the spatial weighting of the HOG features in the K-d tree improves the classification accuracy by over 20%, attaining the best overall rate of 80.90%. This outperforms the Random Forest by about 10%.

The Random Forests achieve a poorer performance on this data set due to its small size. The random selection of samples when constructing the random trees favors the highly represented classes. The K-d tree classifiers are able to overcome this problem, as they use the entire data set. When considering the results at hand, one can conclude that the choice of the classifier strongly depends on the training data set. The Random Forests outperformed the K-d trees on the German Traffic Sign Recognition Benchmark due to the sufficient amount of 26,640 training samples. The K-d tree yields a better performance on the smaller data set containing only 14,763 training images.

5. Feature space reduction

The feature space reduction techniques are designed to minimize the memory and processing requirements as well as yield higher accuracy rates by eliminating less important features. The minimization of resource requirements is particularly important in embedded systems, which are often subject to physical and financial constraints. The feature selection also helps understand the generated features. In the following experiments, the German Traffic Sign Recognition Benchmark and the corresponding HOG 2 feature are used for the evaluation of the Random Forests and Fisher's Criterion for feature space reduction.

5.1. Feature selection

Two selection techniques are implemented for the feature space reduction: Random Forests and Fisher's Criterion. We use the HOG 2 descriptor in the experiments as it yields the best results.

Fisher's Criterion ranks the features according to their ratio of inter-class to intra-class variance. The Fisher Scores of the HOG 2 feature vector are shown in Fig. 5. Each row or column of blocks of this feature is described by 224 values (7 blocks $\times$4 cells $\times$8 bins). Note that the peaks in the variable importance recur every 224 values, coinciding with the interior region of the image. Therefore, the central blocks have a higher variable importance than the marginal ones. This affirms the performance improvement when using the spatial weighting, as the interior of the traffic sign image, containing the pictogram or the speed limit, is more important for the classification than the border regions.

The Random Forest can also be used to evaluate the feature importance based on the variance of the classification errors on the permuted out-of-bag data. The variable importance of the HOG 2 feature values, obtained using a Random Forest of 100 trees, 100 variables and 100 samples, show a similar trend to Fisher's Criterion ranking, where the interior blocks yield a higher score than those on the border.

5.2. Evaluation

The n most important features are selected from the ranking obtained from a Random Forest with 100 trees or Fisher's Criterion. To evaluate the feature space reduction, we combine the feature selection techniques with the Random Forest and SVM classifiers. A Random Forest with 100 trees or an SVM is then trained using this subset.

The overall performance is compared in Table 7. Note that similarly high accuracy rates are obtained by the Random Forest and SVM when using only about one third of the features. The Random Forests outperform the SVMs. The former yield higher classification accuracies on subsets with more than 500 features,

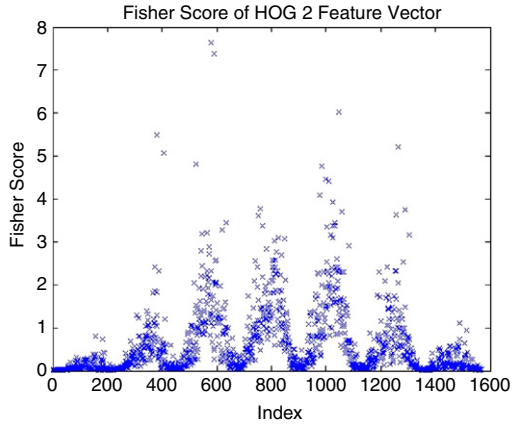


Fig. 5. Feature ranking using Fisher Scores on HOG 2 feature. The peaks in the variable importance recur every 224 values (one row or column of blocks is composed of 7 blocks × 4 cells × 8 bins), coinciding with the interior region of the image.

Table 7
Feature space reduction using Fisher's Criterion and Random Forests. Fisher's Criterion and Random Forest (100 trees, 100 variables and 100 samples) used for feature ranking and classification. Classifiers: Random Forest and linear SVM with $C = 10$.

Feature selection	# Features	Random Forests (%)	Linear SVM (Pegasos) (%)
None	1568	96.69	95.00
Random Forest	500	96.63	96.48
Fisher Criterion	500	96.63	96.41
Random Forest	1000	96.70	95.94
Fisher Criterion	1000	96.85	95.67

due to the random variable selection in the random trees. The accuracy of the SVMs, however, is improved when using only 200–500 feature values.

6. Conclusions and perspectives

A real-time Traffic Sign Recognition system was presented in this paper. The first step of the three stage approach is the image segmentation to reduce the search space. We improve the color enhancement approach proposed by Ruta et al. [3] by introducing an adaptive threshold. In the second stage, the circular and triangular signs are detected using the efficient HOG/linear SVM detector. The combination of these two phases achieves recall and precision rates of over 90% at a processing rate of 18–28 Frames/s.

The candidates found by the detector are identified using multi-class classifiers. We compare the performance of the K-d trees, the Random Forests and the one-vs-all SVM classifiers. We improve the K-d tree accuracy by up to 15% when applying a spatial weighting to focus the Euclidean similarity measure on the interior of the traffic sign. This technique outperforms the K-d tree with the Euclidean, correlation and χ^2 distance metrics on our traffic sign recognition data set with a classification rate about 81%. However, the Random Forest outperforms the K-d tree and SVM on the larger German Traffic Sign Recognition Benchmark, yielding a classification accuracy of 97%. One can conclude that the choice of the suitable classifier depends on the cardinality of the training data set.

Moreover, we employ the Random Forests and Fisher's Criterion feature selection techniques. The benefit of this is two-fold. On the one hand, the feature space dimension is reduced, minimizing the memory and processing requirements. On the other, the classification accuracy of the SVM is improved when using a well-chosen subset of features.

Future work could integrate the temporal information to track the detected traffic signs and reinforce the decision making process. This would further accelerate the candidate detection by restricting the search space. The feature selection can be employed to accelerate the detection phase by reducing the size of the descriptor vectors. Further, this could be combined with other classifiers, such as the Neural Networks. Moreover, the adaptive threshold can be combined with other color enhancements to detect blue traffic signs or green traffic lights, for example.

References

[1] P.G. Jimenez, S. Lafuente-Arroyo, H. Gomez-Moreno, F. Lopez-Ferreras, S. Maldonado-Bascon, Traffic sign shape classification evaluation, part II, FFT applied to the signature of blobs, in: Intelligent Vehicles Symposium, 2005, Proceedings, IEEE, IEEE, 2005, pp. 607–612.
[2] C. Fang, S. Chen, C. Fuh, Road-sign detection and tracking, IEEE Transactions on Vehicular Technology 52 (5) (2003) 1329–1341.
[3] A. Ruta, Y. Li, X. Liu, Real-time traffic sign recognition from video by class-specific discriminative features, Pattern Recognition 43 (1) (2010) 416–430.
[4] S.M. Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, F. Lopez-Ferreras, Road-sign detection and recognition based on support vector machines, IEEE Transactions on Intelligent Transportation Systems 8 (2) (2007) 264–278.
[5] W.-J. Kuo, C.-C. Lin, Two-stage road sign detection and recognition, in: 2007 IEEE International Conference on Multimedia and Expo, 2007, pp. 1427–1430.
[6] A. De La Escalera, L. Moreno, M. Salichs, J. Armingol, Road traffic sign detection and classification, IEEE Transactions on Industrial Electronics 44 (6) (1997) 848–859.
[7] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paetzold, C. Wöhler, Autonomous driving approaches downtown, IEEE Intelligent Systems 13 (6) (1999) 1–14.
[8] F. Moutarde, A. Bargeton, A. Herbin, L. Chanussot, Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular traffic signs recognition system, in: 2007 IEEE Intelligent Vehicles Symposium, 2007, pp. 1122–1126.
[9] A. Ruta, Y. Li, M. Uxbridge, F. Porikli, S. Watanabe, H. Kage, K. Sumi, J. Amagasaki, A new approach for in-vehicle camera traffic sign detection and recognition, in: Proceedings of the IAPR Conference on Machine Vision Applications, Japan, 2009.
[10] N. Barnes, A. Zelinsky, L. Fletcher, Real-time speed sign detection using the radial symmetry detector, IEEE Transactions on Intelligent Transportation Systems 9 (2) (2008) 322–332.
[11] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, N. Shevtsova, Recognition of traffic signs based on their colour and shape features extracted using human vision models, Journal of Visual Communication and Image Representation 17 (4) (2006) 675–685.
[12] B. Alefs, G. Eschemann, H. Ramoser, C. Beleznai, Road sign detection from edge orientation histograms, in: 2007 IEEE Intelligent Vehicles Symposium, June 2007, pp. 993–998.
[13] Y. Xie, L. Liu, C. Li, Y. Qu, Unifying visual saliency with HOG feature learning for traffic sign detection, in: Intelligent Vehicles Symposium, 2009 IEEE, IEEE, 2009, pp. 24–29.
[14] X. Qingsong, S. Juan, L. Tiantian, A detection and recognition method for prohibition traffic signs, in: Image Analysis and Signal Processing (IASP), 2010 International Conference on, IEEE, 2010, pp. 583–586.
[15] I. Creusen, R. Wijnhoven, E. Herbschleb, P. de With, Color exploitation in hog-based traffic sign detection, in: Image Processing (ICIP), 2010 17th IEEE International Conference on, 2010, pp. 2669–2672.
[16] X. Baró, S. Escalera, J. Vitrià, O. Pujol, P. Radeva, Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification, IEEE Transactions on Intelligent Transportation Systems 10 (1) (2009) 113–126.
[17] B. Hoferlin, K. Zimmermann, Towards reliable traffic sign recognition, in: Intelligent Vehicles Symposium, 2009 IEEE, IEEE, 2009, pp. 324–329.
[18] Y.-Y. Nguwi, A. Kouzani, Detection and classification of road signs in natural environments, Neural Computing & Applications 17 (2008) 265–289. http://dx.doi.org/10.1007/s00521-007-0120-z.
[19] A. Bargeton, F. Moutarde, F. Nashashibi, B. Bradai, Improving pan-European speed-limit signs recognition with a new global number segmentation before digit recognition, in: 2008 IEEE Intelligent Vehicles Symposium, 2008.
[20] K. Lim Jr., K. Seng Jr., L. Ang Jr., Intra color-shape classification for traffic sign recognition, in: ICS, 2010, pp. 642–647.
[21] A. Kouzani, Road-sign identification using ensemble learning, in: Intelligent Vehicles Symposium, 2007 IEEE, IEEE, 2007, pp. 438–443.
[22] J. Beis, D. Lowe, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, in: Computer Vision and Pattern Recognition, 1997, Proceedings, 1997 IEEE Computer Society Conference on, IEEE, 2002, pp. 1000–1006.

[23] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The German traffic sign recognition benchmark: a multi-class classification competition, in: International Joint Conference on Neural Networks, 2011.
[24] A. Ruta, F. Porikli, S. Watanabe, Y. Li, In-vehicle camera traffic sign detection and recognition, Machine Vision and Applications (2011) 1–17.
[25] F. Zaklouta, B. Stanciulescu, Warning traffic sign recognition using a hog-based *k*-d tree, in: Intelligent Vehicles, 2011.
[26] F. Zaklouta, B. Stanciulescu, Real-time traffic sign recognition using spatially weighted hog trees, in: IEEE International Conference on Advanced Robotics, ICAR, IEEE, 2011.
[27] F. Zaklouta, B. Stanciulescu, O. Hadmoun, Traffic sign classification using *k*-d trees and random forests, in: International Joint Conference on Neural Networks, 2011.
[28] D. Ciresan, U. Meier, J. Masci, J. Schmidhuber, A committee of neural networks for traffic sign classification, in: International Joint Conference on Neural Networks, 2011.
[29] P. Sermanet, Y. LeCun, Traffic sign recognition with multi-scale convolutional networks, in: International Joint Conference on Neural Networks, 2011.
[30] L. Breiman, Random Forests, Machine Learning 45 (1) (2001) 5–32.
[31] T. Ho, Random decision forests, in: ICDAR, IEEE Computer Society, 1995, p. 278.
[32] S. Shalev-Shwartz, Y. Singer, N. Srebro, Pegasos: primal estimated sub-gradient solver for svm, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 807–814.

Fatin Zaklouta finished her Ph.D. Thesis on Multi-class Object Recognition for ADAS and Video-surveillance in December 2011 at the Robotics Laboratory of the Mines-ParisTech/Armines. She obtained her M.Sc. at the University of Passau, Germany in 2008. Her main research interests are computer vision and machine learning for object detection and pattern recognition.

Bogdan Stanciulescu is a research scientist at the Robotics Laboratory of the Mines-ParisTech/Armines since 2003. He was a post-doc fellow at LAMIH – Valenciennes in 2002–2003. He defended his Ph.D. Thesis in 2002 at ENSTA – Paris, in the field of imaging, computer vision and robotics. His main research interests are in image processing and computer vision, machine learning techniques for object detection and pattern recognition, augmented reality with applications in advanced driver assistance systems and intelligent video-surveillance systems. He is currently working on object detection topics in ITS applications, such as pedestrian and car detection, road markings detection, people tracking and image restoration.