

# Real-time traffic sign recognition using spatially weighted HOG trees

Fatin Zaklouta and Bogdan Stanciulescu

**Abstract**—Traffic sign recognition is one of the main components of a Driver Assistance System (DAS). This paper presents a real-time traffic sign recognition system. It consists of three stages: 1) an image segmentation using red color enhancement to reduce the search space, 2) a HOG-based Support Vector Machine (SVM) detection to extract the traffic signs, and 3) a tree classifier (K-d tree or Random Forests) to identify the signs found. This methodology is tested on images under bad weather conditions and poor illumination. The tree classifiers achieve high classification rates for the German Traffic Sign Recognition Benchmark and the ETH 80 dataset. The K-d tree classification is improved by introducing a Gaussian spatial weighting to favor the interior blocks of the HOG descriptors.

## I. INTRODUCTION

Traffic sign recognition is an important part of a Driver Assistance System (DAS). Traffic signs enhance safety by informing the driver of speed limits, warning him against possible dangers such as slippery roads, imminent road works or pedestrian crossings. Their bright colors and simplified pictograms make them easily comprehensible and perceivable.

Three of the main challenges facing traffic sign detection systems are i) the poor image quality due to low resolution, bad weather conditions or poor illumination, ii) the rotation, occlusion and deterioration of the signs, iii) the need for a response in real-time in a DAS.

In this paper, we propose a real-time HOG-based traffic sign detection combined with an image segmentation. The detected candidates are passed on to a tree classifier to determine the nature of the sign. The rest of this paper is organized as follows: Section II gives an overview of the state of the art of traffic recognition systems. Section III briefly describes our approach. Section IV outlines the detection phase and the segmentation technique used. Section VI describes the tree classifiers. The evaluation and results of the proposed methodology are given in Section VII. A brief outlook on future improvements is given in Section VIII.

## II. STATE OF THE ART

Traffic recognition algorithms divide the problem into two parts: 1) the detection and 2) the classification.

The detection is often based on the color segmentation of the image to retrieve the Regions of Interest (ROI). Since the direct thresholding of the RGB channels is sensitive to changes in illumination, the relation between the RGB colors is used. In [18], the color enhancement is used to extract

red, blue and yellow blobs. This transform emphasizes the pixels where the given color channel is dominant over the other two in the RGB color space. The RGB color Haar wavelets are used in [2]. In [15], chromatic and achromatic filters are used to extract the red rims and the interior white of the traffic signs respectively. The HSI color model (Hue Saturation Intensity) is used in [17] as it is invariant to illumination changes. It is pointed out in [9], however, that HSI is computationally expensive due to its nonlinear formulae.

Other 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. Franke et al. [11] use Distance Transform (DT) template matching to detect circular and triangular signs. Similarly, Ruta et al. [18] use the Color Distance Transform, where a separate distance transform DT is computed for every color channel. The advantage of matching DTs over edge images is that the similarity measure is smoother. It is, however, sensitive to rotations and occlusions. In [15], an SVM is trained on the Distance to Border (DtB) vectors to classify the shape of a detected traffic sign. In [13], the FFT of candidate signs are compared to those of the templates. This feature is robust to rotation, deformation and scaling. The circular Hough transform, used in [16], [19] for detecting speed limits, and the simpler Radial Symmetry Detector[3] are restricted to circular traffic signs.

Many recent approaches use gradient orientation information in the classification phase. Gao et al. [12] classify the candidate traffic signs by comparing their local edge orientations at arbitrary fixation points with those of the templates. In [1], Edge Orientation Histograms are computed over class-specific subregions of the image. The Histogram of Oriented Gradients (HOG) [8], initially used for pedestrian detection, has been adapted to traffic sign detection in several works. In [23], [17], the Regions of Interest (ROI) obtained from color-based segmentation are classified using a HOG-based classifier. To integrate color information in the HOG descriptor, Creusen et al [7] 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.

## III. SYSTEM OVERVIEW

In this paper, we evaluate the segmentation based on the color enhancement proposed in [18]. The resulting binary masks are used to extract traffic sign ROIs. The candidate

F. Zaklouta and Bogdan Stanciulescu are with Robotic Center at the Mines ParisTech, Paris, France  
fatin.zaklouta@ensmp.fr  
bogdan.stanciulescu@ensmp.fr

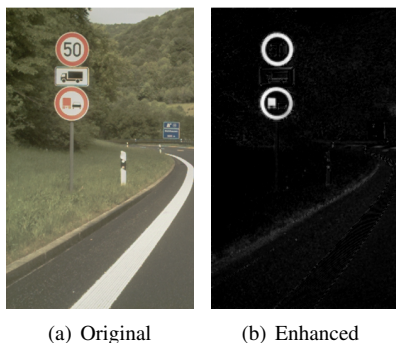


Fig. 1. Image segmentation

set is then refined using a HOG-based SVM detector. The signs found are identified using a tree classifier.

#### IV. DETECTION

##### A. Filtering Mask

Color Enhancement is used by Ruta et al. in [18]. However, the subsequent stages in their proposal are computationally expensive. They exploit the a priori knowledge of the red rim color of the traffic signs to determine the Regions of Interest (ROI). The ROIs are found using the time consuming region growing method and classified by matching their Color Distance Transforms to a template. This matching is not applicable to occluded or rotated signs.

In this paper, we adapt the color enhancement stage and improve the classification step using a HOG-based SVM detector. Both the search space and the number of false alarms are reduced when applying the binary mask obtained from the red color enhancement.

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

$$f_R(x) = \max(0, \min(x_R - x_G, x_R - x_B)/s),$$

where  $s = x_R + x_G + x_B$ . The pixels having a dominant red component are extracted, while all others are set to zero. The resulting enhanced image is thresholded. Fig. 1b shows the result of the red color channel enhancement. Note that the rims and the red truck pictogram are emphasized.

The result  $f(x)$  is thresholded using  $f(x) > \mu + 4\sigma$ , where  $\mu$  and  $\sigma$  are the mean and standard deviation of the pixel values over the entire image. The global mean helps take the illumination of the whole image into consideration, while a mean computed locally is more sensitive to small illumination variations in the image.

##### V. HOG-BASED SVM DETECTION

The Histogram of Oriented Gradients, proposed by Dalal and Triggs [8] for pedestrian detection, is also popular in the recognition of different types of objects. We choose to use this feature for the traffic sign detection due to its scale-invariance, local contrast normalization, coarse spatial sampling and fine weighted orientation binning. The unsigned gradients allow for the detection of both static signs as well as dynamic, illuminated signs with a single detector. This

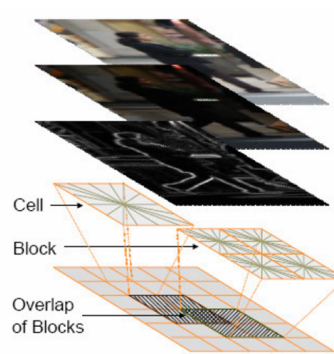


Fig. 2. Structure of HOG descriptor [8]

is not possible with other features, such as Haar wavelet features [22], where the difference between the sums of pixels in two regions indicates the direction of the gradient.

Fig. 2 illustrates the structure of the HOG descriptor used for pedestrian detection. The image of a traffic sign is divided into overlapping blocks. Each block, in turn, is divided into non-overlapping cells. The gradient magnitude and angle are computed for each pixel. A histogram of the orientations of the gradients of each cell is formed. The magnitude of the gradient is used as a vote weight. The histograms of the cells of each block are concatenated to form the descriptor. We test HOG on both grayscale and color images. For the latter, as in [8], the separate gradients are calculated for each color channel. The one with the largest norm is taken as the pixel's gradient vector.

In this paper, we use a 144 value descriptor with 9-bin histograms for all our experiments. The sign/non-sign training images are rescaled to 16x16 pixel images. Their HOG descriptors are computed and used to train two linear SVM classifiers (one for round signs and another for triangular ones) using the SVMLight<sup>1</sup> library. The  $n$  resulting support vectors are combined to a single global vector.

$$v = \sum_i^n \alpha_i \cdot v_i,$$

where  $\alpha_i$  is the confidence in support vector  $v_i$ . It is computed using the classification error. This reduction to one support vector accelerates the detection as the HOG descriptors of the candidates are compared to a single vector instead of several. 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 passed on to the tree classifier in the classification phase.

#### VI. CLASSIFICATION

##### A. K-d Trees

A K-d tree is a binary search tree organizing  $k$ -dimensional data points. Each non-leaf node splits the data into two subspaces in the feature  $i$  with the highest variance at that level. Therefore, each

<sup>1</sup><http://svmlight.joachims.org/>

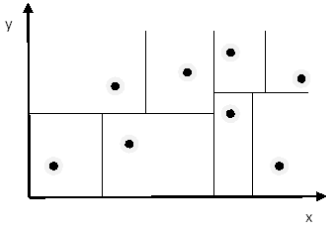


Fig. 3. Example of 2D K-d tree

node contains a  $(k - 1)$ -dimensional hyperplane  $H_i = \{x_1, \dots, x_{i-1}, i, x_{i+1}, x_k\}$ ,  $1 \leq i \leq k$  parallel to the  $i$ -th axis. In Fig. 3, the 2-D space is divided alternately by 1-D lines parallel to the  $x$  and the  $y$  axes. To ensure that the tree is balanced, the median value  $v_i$  of the  $i$ -th dimension is used for the split in each node. The left subtree contains data points smaller than  $v_i$  and the right subtree points larger than  $v_i$ . This division is repeated until each data point is represented by a leaf.

The K-d tree is a nearest-neighbor-based search tree. We use the Best Bin First algorithm [4] to perform an approximate Nearest Neighbor search. This efficient variant of the K-d tree search algorithm allows for the indexing of higher dimensional spaces which is required when using HOG vectors or other large descriptors. Instead of searching the entire tree in an exhaustive manner, Beis et Lowe [4] propose to search the nodes closest to the query sample using a priority queue. This queue contains candidate nodes ranked according to their distance to the query. The ranking determines the order in which the nodes are examined. During the search, the siblings of the current node being examined are iteratively added to the priority queue. The search is terminated when the algorithm scans a predefined maximum number of nodes  $E_{\max}$ .

A nearest neighbor approach is more suitable for the challenge dataset due to the large variation in the number of training samples of the different classes. Training a learning-based classifier such as an SVM on an imbalanced dataset often requires parameter fine-tuning, making the system less generic.

During the testing phase, the  $k$  Nearest Neighbors ( $kNN$ ) are retrieved for each candidate. The vote of the class, to which each of the nearest neighbors belongs is incremented. The vote is weighted with the reciprocal of the Euclidean distance to the candidate. The maximum vote determines the class of the candidate.

$$\text{vote}_{\text{class}} = \sum_1^k \frac{1}{\text{distance}(kNN, \text{candidate})}, kNN \in \text{class} \quad (1)$$

**Spatial Feature Weighting** We further introduce a spatial weighting of the descriptors, in the form of a 2-dimensional Gaussian, to concentrate the approximate nearest neighbor search on the interior HOG blocks of the image. When computing the 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.

## B. Random Forest

Random Forests were introduced by Breiman and Cutler in [6]. An ensemble of random trees forms a Random Forest. To classify a sample, the classification of each random tree in the forest is taken into account. The class label of the sample is the one with the majority of the votes.

A random tree is grown as follows:

- A subset of  $n < N$  training samples is randomly chosen with replacement from the original training set containing  $N$  samples. The tree is grown using this subset and is not pruned.
- A subset of  $m < M$  features is randomly chosen at each node. The best split at this node is computed using this subset.

The random forests achieve state-of-the-art performance in many multi-class classification applications. In [5], the random forests outperform the SVM in classifying images from the Caltech-101 and Caltech-256 data sets. Khoshgof-taar et al. show in [14] that random forests perform well on binary classification problems with imbalanced data sets and outperform SVM, Naives Bayes,  $kNN$  and C4.5 classifiers. A further advantage is that they are fast to build, easy to implement in a distributed computing environment and that they enable online learning.

## VII. RESULTS

To evaluate the proposed detection algorithm and the performance of the color segmentation as a filter, we use image sequences acquired in both urban and highway environments under different meteorological conditions. A predicted bounding box is considered correct if it overlaps more than 50% of the ground truth. The evaluation is based on ROC curves as well as the recall and precision values defined as follows:

$$\text{recall} = \frac{\text{true positives detected (TP)}}{\text{total true positives}}$$

$$\text{precision} = \frac{\text{true positives detected (TP)}}{\text{all detections}}$$

We evaluate the tree classifiers on both our data set as well as the German Traffic Sign Recognition Benchmark, which contains a significantly larger number of classes and images. We also evaluate the generalization capability of the tree classifiers using the ETH 80 data set containing 3280 images of 80 different objects from 8 classes.

### A. Data set

Our data set contains 24 classes of traffic signs: 15 round and 9 triangular. The train and test data sets contain 14765 and 1584 signs respectively. There is a significant imbalance in the number of training samples for each class. The size of the training sets varies from 15 to 3852 images. This imbalance favors hierarchical classification approaches, such as trees, over machine learning techniques, such as an SVM.

TABLE I

EFFECT OF SEGMENTATION MASKS ON DETECTION PERFORMANCE

Mask	Grayscale HOG			RGB HOG		
	Recall	Precision	Time	Recall	Precision	Time
None	75.82	90.16	36 ms	84.72	89.89	49 ms
Red	84.66	94.77	49 ms	90.21	90.90	56 ms

### B. Detection

The evaluation of the performance with and without the red color enhancement is illustrated in Fig. 4. We also compare the performance of other segmentation techniques based on the tophat and the bottom hat morphological filters as well as the chromatic filter used in [15] for traffic sign recognition. The same threshold is used for all four filters to obtain binary masks.

All four approaches significantly reduce the number of false positives. The morphological operators reduce the hit rate by 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.

Table I shows the effect of using the segmentation mask on the detection performance. The red channel color enhancement improves both the recall and the precision of the HOG detector. It achieves the best results with a recall of 90.21% at a precision of 90.9%. One can also note that the HOG descriptors computed on the RGB color space achieve a 10% higher recall than the HOG computed on the grayscale images. This improvement is due to the strong gradients in the red color channel which are not as intense in the gray images.

Table I also shows the average time (in ms) required to process a 752x480 pixel frame on a 2.93 GHz Intel Core i7 computer with a C++/OpenCV implementation. The grayscale HOG detector achieves real-time performance, running at 27.7 frames per second. The improved red enhancement segmentation mask requires a longer processing time, yet can run in near real-time at 20.4 frames per second.

### C. Classification

In the classification phase, the K-d tree and the Random Forests are built using the training samples. For each candidate from the test set, the  $k$  nearest neighbors in the K-d tree are found. The class vote is inversely proportional to the distance between the candidate and its closest match (see Section VI-A). The  $E_{max}$  parameter is set to 5000. As for the Random Forests, each tree in the ensemble contributes to the class vote of the test sample. A minimum of 100 samples and a subset of 100 variables are used to construct each tree. The forest consists of 500 trees. These parameters were chosen empirically. Their effect on the classification result is shown in Section VII-D.

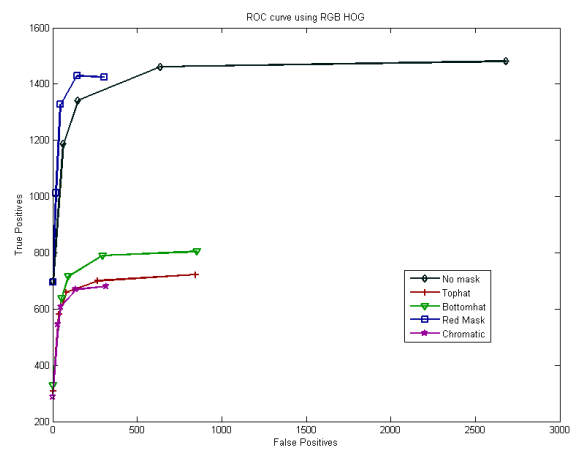
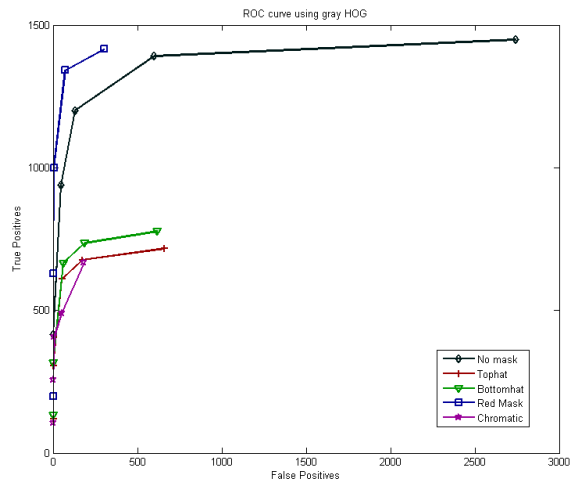


Fig. 4. Color segmentation using grayscale and color HOG

TABLE II  
PARAMETERS OF THE HOG DESCRIPTORS

Dimension	Cell	Block	Stride	K-d Tree	Random Forest
144	4x4	4x4	4x4	63.01 %	62.31 %
441	2x2	4x4	4x4	78.28 %	80.05 %
1764	2x2	4x4	2x2	81.38 %	83.46 %

We test different size HOG descriptors to better analyze the performance of the tree classifiers. The results are shown in Table II. The Random Forests outperform the K-d trees when using the larger HOG descriptors. The random selection of variables and their large number in the ensemble make them more robust and performant than the single K-d tree.

### D. German Traffic Sign Recognition 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 [20]. This image data set contains 43 classes, 26,640 train images and 12,569 test images.

The three different HOG features used are listed in Table III. The block and cell sizes are measured in pixels. All

TABLE III  
PARAMETERS OF THE HOG DESCRIPTORS

Name	Dimension	Cell	Block	Stride	Bins	Semicircle
HOG 1	1568	5x5	10x10	5x5	8	true
HOG 2	1568	5x5	10x10	5x5	8	false
HOG 3	2916	4x4	8x8	4x4	9	true

TABLE IV  
ACCURACY RATES WHEN VARYING  $E_{\max}$  IN K-D TREE

$E_{\max}$	HOG 1	HOG 2	HOG 3
500	71,30 %	68,84 %	71,08 %
1000	73,03 %	70,40 %	72,75 %
1500	73,76 %	71,25 %	73,65 %
2000	73,94 %	71,87 %	74,00 %
2500	74,37 %	72,03 %	74,53 %
5000	74,92 %	73,39 %	75,03 %

the images used are resized to 40x40 pixels using a bilinear interpolation. The precalculated descriptors for both the train and test images are available on the benchmark website.

1) **K-d tree:** The construction and the parameters of the K-d tree are described in Section VI-A. Five nearest neighbors are retrieved for each test candidate, i.e.  $k = 5$ . This value was determined empirically. The performance of the K-d tree also depends on the value of the  $E_{\max}$  parameter. Its effect on the classification results is shown in Table IV. Increasing the  $E_{\max}$  parameter by a factor of 10 increases the classification rate by up to 4.55%. The  $E_{\max}$  parameter in the K-d tree is set to 5000 in further experiments, since it achieves the best results for all four HOG descriptor sizes.

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

The 30, 50 and 80 speed limit signs are most frequently confused. 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. With  $E_{\max} = 5000$ , the accuracies of the HOG 1, 2 and 3 descriptors are increased to 88.53%, 88.73% and 90.39% respectively.

2) **Random Forests:** To evaluate the performance of the Random Forest classifier, we use the different sized HOG descriptors and vary its parameters: the number of trees in the forest, the number of features to be chosen at random and the size of the training sample subset. The structure of the Random Forests and their parameters are explained in Section VI-B. The results of changing the parameters are shown in Table V. The HOG 2 descriptor is used. The performance of the Random Forests does not vary significantly (0 to 2%) when the parameters are changed,

TABLE V  
ACCURACY RATES WHEN VARYING THE PARAMETERS OF THE RANDOM FORESTS USING THE HOG 2 DESCRIPTOR

	Nb samples	Nb Features	Nb Trees	Classification Rate
Features	100	10	500	95.5 %
	100	50	500	97.1 %
	100	75	500	97.0 %
	100	100	500	97.1 %
Samples	10	100	500	97.1 %
	100	100	500	<b>97.2 %</b>
	500	100	500	95.2 %
Trees	100	100	50	96.0 %
	100	100	100	96.7 %
	100	100	300	<b>97.2 %</b>
	100	100	500	97.1 %
	100	100	750	<b>97.2 %</b>

which makes them more generic and easier to use.

The Random Forests achieve 95.1%, 97.2% and 95.2% for the HOG 1, 2 and 3 descriptors respectively. The number of trees is set to 500, the number of features and samples to 100. The HOG 2 descriptor perform slightly better than the other three. The Random Forests achieve about 7% higher classification rates than the K-d trees with spatially weighted blocks when using the HOG 1, 2 and 3 descriptors. 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 of the similarity measure caused by the background is less significant. The randomness of the Random Forest classifier makes it less sensitive to the variations than the K-d tree, which uses the entire descriptor set.

#### E. ETH 80

Due to their standardisation, traffic signs show a low intra-class variability. They may be considered "easy" to classify using a Nearest Neighbor approach since they are all almost identical. Therefore, we also evaluated the tree classifiers using the ETH-80 dataset, first introduced by Eichhorn and Chapelle [10]. This dataset consists of images of 8 classes, with 10 objects each. Each object was photographed at 41 different angles i.e. a total of 3280 images. In their paper, Eichhorn and Chapelle use 8 one-vs-rest SVMs to classify the SIFT and JET features. They evaluate their classifiers using a leave-one-object-out crossvalidation. The same experiment was repeated by Suard et al [21] using HOG and graph features with 28 one-vs-one SVMs. The HOG features used consist of 8464 values: 96x96 pixel images, 4x4 pixel cells, 8x8 pixel blocks and a 4x4 pixel block stride.

We evaluate the tree classifiers using the same HOG features as in [21]. We also train one-vs-one SVMs with smaller HOG descriptors with 784 values: 64x64 pixel images, 8x8 pixel cells, 16x16 pixel blocks and a 8x8 pixel block stride. The average results of the leave-one-object-out crossvalidation of all the approaches are shown in Table VI. A single K-d tree achieves a comparable classification rate to that of the 28 one-vs-one SVMs using the same sized HOG descriptors. The SVM performs poorly using the

TABLE VI  
ETH 80 CLASSIFICATION RESULTS

Source	Feature	Classifier	Accuracy
[10]	SIFT	SVM	74%
[21]	HOG (8464)	SVM	<b>90.1%</b>
[21]	Graph	SVM	78%
[21]	HOG (8464) + Graph	SVM	<b>94.1%</b>
This paper	HOG (8464)	K-d trees	87.99%
This paper	HOG (784)	SVM	43.78%
This paper	HOG (784)	K-d trees	<b>90.46%</b>
This paper	HOG (784)	Random Forests	86.49%

HOG descriptor with the coarser spatial sampling and only 784 values. The high classification rate of the K-d trees, however, did not suffer. Therefore, the K-d tree approach is less time and memory consuming for the training and testing phases as well as with respect to the features used. The spatial weighting of the blocks achieved 89.1% and 85.5% for the 784 and 8464 valued HOG descriptors respectively. The prioritization of the interior blocks did not improve the results due to the irregular shapes of the objects.

## VIII. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

In this paper, a real-time traffic sign recognition system is presented. The color enhancement segmentation technique is evaluated and used to improve the recall and precision of the HOG-based SVM detector. It achieves recall and precision rates of 90%.

The performance of the K-d tree and the Random Forests is evaluated both on our data set and the publicly available German Traffic Sign Recognition Benchmark [20]. The Random Forests outperform the K-d trees, achieving a classification rate of 97.2%. Further, we introduce a spatial Gaussian weighting to prioritize the interior HOG blocks in the Nearest Neighbor classification. This increases the K-d tree performance by 15%.

We evaluate the generalization capability of the tree classifiers using the ETH 80 dataset, achieving state-of-the-art classification rates of 90% and outperforming the SVM classifiers when using a coarser spatial sampling.

### B. Future Works

Future works include enriching our traffic sign data set. The detection rates can be further improved by using spatial and temporal information i.e. restricting the search space to the sides of the road and reinforcing the classification decision over several consecutive frames. The classification rates can be augmented by exploiting other visual features such as color histograms and distance transforms.

## REFERENCES

- [1] B. Alefs, G. Eschemann, H. Ramoser, and C. Beleznaï. Road sign detection from edge orientation histograms. In *2007 IEEE Intelligent Vehicles Symposium*, pages 993–998, June 2007.
- [2] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In *Proceedings of the IEEE Symposium on Intelligent Vehicles*, pages 255–260. Citeseer, 2005.
- [3] N. Barnes, A. Zelinsky, and L.S. Fletcher. Real-time speed sign detection using the radial symmetry detector. *Intelligent Transportation Systems, IEEE Transactions on*, 9(2):322–332, 2008.
- [4] J.S. Beis and D.G. 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*, pages 1000–1006. IEEE, 2002.
- [5] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. ICCV*, 2007.
- [6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [7] I.M. Creusen, R.G.J. Wijnhoven, E. Herbschleb, and P.H.N. de With. Color exploitation in hog-based traffic sign detection. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2669–2672, 2010.
- [8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l’Europe, Montbonnot-38334, June 2005.
- [9] A. De La Escalera, L.E. Moreno, M.A. Salichs, and J.M. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6):848–859, 1997.
- [10] J. Eichhorn and O. Chapelle. Object categorization with SVM: kernels for local features. *MPIK*, 2004.
- [11] U. Franke, D. Gavrilu, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler. Autonomous driving approaches downtown. *IEEE Intelligent Systems*, 13(6):1–14, 1999.
- [12] X.W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and 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):675–685, 2006.
- [13] P. Gil-Jimenez, S. Lafuente-Arroyo, H. Gomez-Moreno, F. Lopez-Ferreras, and S. Maldonado-Bascon. Traffic sign shape classification evaluation. Part II. FFT applied to the signature of blobs. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 607–612. IEEE, 2005.
- [14] T.M. Khoshgoftaar, M. Golawala, and J. Van Hulse. An empirical study of learning from imbalanced data using random forest. In *19th IEEE International Conference on Tools with Artificial Intelligence, 2007. ICTAI 2007*, volume 2, pages 310–317, 2007.
- [15] 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. *IEEE transactions on intelligent transportation systems*, 8(2):264–278, 2007.
- [16] F. Moutarde, A. Bargeton, A. Herbin, and 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*, pages 1122–1126, 2007.
- [17] X. Qingsong, S. Juan, and L. Tiantian. A detection and recognition method for prohibition traffic signs. In *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, pages 583–586. IEEE, 2010.
- [18] A. Ruta, Y. Li, and X. Liu. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition*, 43(1):416–430, 2010.
- [19] A. Ruta, Y. Li, M. Uxbridge, F. Porikli, S. Watanabe, H. Kage, K. Sumi, and J. Amagasaki. A New Approach for In-Vehicle Camera Traffic Sign Detection and Recognition. In *Proc. IAPR Conference on Machine Vision Applications, Japan*, 2009.
- [20] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks*, 2011.
- [21] F. Suard, A. Rakotomamonjy, and A. Benshair. Object Categorization using Kernels combining Graphs and Histograms of Gradients. *Lecture Notes in Computer Science*, 4142:23, 2006.
- [22] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, volume 1, pages 511–518, 2001.
- [23] Y. Xie, L.F. Liu, C.H. Li, and Y.Y. Qu. Unifying visual saliency with HOG feature learning for traffic sign detection. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 24–29. IEEE, 2009.