Traffic Sign Classification using K-d trees and Random Forests

Fatin Zaklouta and Bogdan Stanciulescu and Omar Hamdoun

Abstract—**In this paper, we evaluate the performance of K-d trees and Random Forests for traffic sign classification using different size Histogram of Oriented Gradients (HOG) descriptors and Distance Transforms. We use the German Traffic Sign Benchmark data set [1] containing 43 classes and more than 50,000 images. The K-d tree is fast to build and search in. We combine the tree classifiers with the HOG descriptors as well as the Distance Transforms and achieve classification rates of up to 97% and 81.8% respectively.**

I. Introduction

Traffic sign recognition is an essential component of a Driver Assistance System (DAS), providing drivers with safety and precaution information. Traditionally, traffic sign recognition systems consist of two phases: detection and classification. In the scope of our previous work, we developed a Histogram of Oriented Gradients (HOG) based Support Vector Machines (SVM) detector to find traffic sign candidates in urban and highway environments under varying illumination and weather conditions. In this paper, we evaluate the techniques used in the classification phase. We compare the performance of K-d trees and Random Forests using HOG descriptors and Distance Transforms computed on images from the German Traffic Sign Benchmark data set [1].

The traffic signs are designed to be easily noticeable, with high contrast and bright colors. Simple pictograms and characters are used to make them easily comprehensible. We exploit this design aspect when using the Histogram of Oriented Gradients and the Distance Transform images. The standardization of these traffic signs favors a nearest neighbor approach, which we implement efficiently using tree structures. One advantage of using trees is that they are fast and easy to build, update and search in. Another is that they cope well with imbalanced data sets, whereas other learning algorithms like SVM and Neural Networks require parameter fine tuning.

The state of the art on traffic sign recognition is described in Section II. The features used for training the tree classifiers are described in Section III. Sections IV and V describe the construction and the parameters of the tree classifiers as well as the corresponding efficient search algorithms. The experiments performed to evaluate the features and the tree classifiers are described in Section VI.

II. Related Work

Many approaches for traffic sign recognition have been proposed in the past two decades. These are mainly based on color and shape features.

Fatin Zaklouta, Bogdan Stanciulescu and Omar Hamdoun are with the Robotics Center, Mines ParisTech, Paris, France (email: {fatin.zaklouta, bogdan.stanciulescu,omar.hamdoun}@mines-paristech.fr).

Color-based methods threshold the image to obtain the regions of interest (ROI). In [2], De La Escalera et al use the intensity ratio of the values of one color channel, e.g. red, to the sum of all the RGB components to threshold the image. The resulting regions are classified using a Neural Network. In [3], color-sensitive Haar wavelet features are combined with Adaboost to detect and track traffic signs. More complex approaches include CIECAM97 [4], which is a color appearance standard based on the human perception. The gray-scale based approaches are, however, more robust to changes in illumination than the color-based ones.

Shape-based methods exploit the invariance and symmetry of the traffic signs in the detection phase. In [5], a K-d tree is built with gradient and shape descriptors and used to recognize traffic signs. Franke et al. [6], [7] use Distance Transform (DT) template matching to classify circular and triangular signs. The advantage of matching DTs over edge images is that the similarity measure is smoother. Similarly, Ruta et al. [8] use the Color Distance Transform, where a separate distance transform DT is computed for each color channel. The classification is performed using a nearest neighbor template matching approach. In [9], an SVM is trained on the Distance to Border (DtB) vectors to classify the shape of a detected traffic sign. In [10], the FFT signatures of candidate signs are compared to those of the templates. This feature is robust to rotation, deformation and scaling.

Many recent approaches use gradient orientation information to detect and classify traffic signs. In [11], Edge Orientation Histograms are computed over class-specific subregions of the image. The Histogram of Oriented Gradients (HOG) [12], initially used for pedestrian detection, have been adapted to traffic sign detection in several works. In [13], [14], the Regions of Interest obtained from color-based segmentation are classified using a HOG-based classifier. To integrate color information in the HOG descriptor, Creusen et al [15] concatenate the HOG descriptors calculated on each of the color channels.

Trees are a popular alternative to learning approaches such as SVM and Neural Networks since they are more performant when using imbalanced data sets. A further advantage is that they are fast to build and easy to update. The K-d tree is a nearest-neighbor-based search tree. It is combined with an Radial Basis Neural Network in [5] to classify traffic signs using gradient and shape descriptors. The Best Bin First algorithm [16] is an efficient variant of the K-d tree search algorithm which allows for the indexing of higher dimensional spaces. Random Forests achieve state-of-the-art performance in classifying Caltech-101 and Caltech-256 objects in [17]. It is shown in [18] that they also outperform

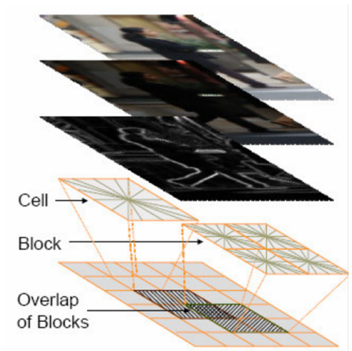

Fig. 1. Structure of HOG descriptor [12]

Fig. 2. Comparison of Distance Transforms (DT) using bottom hat thresholding and Canny edge detection. Left: original image, center: bottom hat threshold result and corresponding DT, right: Canny edge detection and corresponding DT

various other classifiers such as SVM and the Naives Bayes classifier.

III. Features

- **Histogram of Oriented Gradients**
  The Histogram of Oriented Gradients, proposed by Dalal and Triggs [12] for pedestrian detection, has become popular for the recognition of different types of objects. Fig. 1 illustrates the structure of the HOG descriptor. The image of a traffic sign is divided into overlapping blocks. Each block, in turn, is divided into non-overlapping cells. The gradient orientation and magnitude are computed for each pixel. A histogram of these orientations is formed for each cell. The magnitude of the gradient is used as a vote weight. The histograms of the cells of each block are concatenated to form the HOG descriptor.
- **Distance Transforms**
  Distance Transforms (DT) are an efficient method to determine the similarity between two contour images. Different metrics can be used. In our case, the pixel value in the DT is the Euclidean distance of this pixel to the nearest nonzero pixel in the binary image.

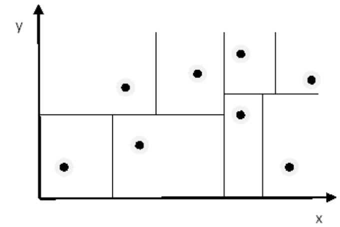

Fig. 3. Example of 2D K-d tree

We use the black hat transform with a filter size of 2x2 pixels and an adaptive threshold to obtain the binary image. For each image, we apply the blackhat (bottom-hat) transform. The blackhat transform is defined as the difference between the closing of the image and the image itself. This morphological filter emphasizes dark pixels with a high contrast to their local environment, which is the case for the interior of the traffic signs. The resulting image is then thresholded. This adaptive threshold corresponds to the mean pixel value over the whole image. This helps take the global illumination into consideration, while a local mean would be more sensitive to small illumination variations in the image. The result is a segmentation of the dark regions that are surrounded by light pixels: for example pictograms and characters. Fig. 2 shows the result of the bottom hat thresholding, the Canny edge detection and their corresponding DT. Note that the bottom hat thresholding preserves the details of the pictograms. The Canny edge detector uses the first derivative of a Gaussian. It considers the weak gradients around the characters in a poorly illuminated image as noise. On the other hand, the bottom hat operator considers local gradients and is less sensative to global variations in illumination. The experiments in Section VI show that the bottom hat based distance threshold yield higher classification rates than the Canny edge based one.

IV. 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 node contains a $(k-1)$-dimensional hyperplane $H_i = \{x_1, \ldots, x_{i-1}, i, x_{i+1}, x_k\}, 1 \leq i \leq k$ parallel to the i-th axis. In Fig. 3, the 2-D space is divided alternatingly 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 [16] 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 [16] 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 neighbours 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.

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

V. Random Forest

Random Forests were introduced by Breiman and Cutler [19]. 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 [17], the random forests outperform the SVM in classifying images from the Caltech-101 and Caltech-256 data sets. Khoshgoftaar et al. show in [18] 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 they enable online learning.

VI. Results and Analysis

Within the scope of the German Traffic Sign Recognition Competition [1], a data set containing 43 classes, 26,640 train images and 12,569 test images are provided. The precalculated HOG 1, 2 and 3 for both the train and test images are also made available. The HOG 4 descriptors are computed within the bounding box of the traffic sign, i.e. not including the 10% border. Fig. 4 shows an example image and its bounding box.

Fig. 4. Bounding Box of traffic sign image

TABLE I
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
HOG 4	2592	4x4	8x8	4x4	8	true

To evaluate the performance of the K-d trees and Random Forests, we use four HOG descriptors of sizes 1568, 1568, 2916 and 2592. The dimensions used to compute these descriptors are listed in Table I. The block and cell sizes are measured in pixels. All the images used are resized to 40x40 pixels using a bilinear interpolation.

We also use the Distance Transform images. Again, the images in the bounding box are used. These are resized to 50x50 pixels. We test both the entire resized image and an interior patch of 41x41 pixels, further eliminating the background and capturing only the pictogram or characters inside the traffic sign.

The parameters of the two classifiers were also varied to examine their effect on the classification performance.

A. K-d trees

TABLE II
Classification Results when varying E_{MAX} in K-d tree

E_{max}	HOG 1	HOG 2	HOG 3	HOG 4
500	71,30 %	68,84 %	71,08 %	91.3 %
1000	73,03 %	70,40 %	72,75 %	92.1 %
1500	73,76 %	71,25 %	73,65 %	92.4 %
2000	73,94 %	71,87 %	74,00 %	92.6 %
2500	74,37 %	72,03 %	74,53 %	92.5 %
5000	74,92 %	73,39 %	75,03 %	92.9 %

The contruction and the parameters of the K-d tree are described in Section IV. 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 depends not only on the features used, but also on the value of the E_{max} parameter. Its effect on the classification results is shown in Table II. 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° to 360°, while the two latter use unsigned gradients i.e. 0° to 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. HOG 4 performs the best with a classification rate of 92.9%. It not only has a fine spatial binning but is also computed on the bounding box rather than including the 10% border i.e. taking less background information into account. The classification rate is improved by about 20% when using the bounding box of the image. This shows the extent at which the varying background in the 10% around the traffic sign perturbates the similarity measurement and the nearest neighbor approach.

TABLE III
CLASSIFICATION RESULTS OF DISTANCE TRANSFORM FEATURES USING K-D TREE

Feature	Bottom hat	Canny
DT 50x50	51.5 %	50.4 %
DT 41x41	67.1 %	63.1 %

To further evaluate the effect of the features used on the performance of the K-d tree, we also use the Distance Transforms. The results obtained are shown in Table III. The HOG descriptors achieve higher classification results than the DT. Eliminating the background information leads to a higher comparison precision in the K-d tree and a better performance. The classification rate is increased by 16% in the bottom hat and 13% in the Canny edge based DT when using the interior patch rather than the entire image. The bottom hat thresholding approach achieves higher classification rates than that using the Canny edge detection.

B. Random Forests

TABLE IV
CLASSIFICATION RESULTS 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	97.2 %
	500	100	500	95.2 %
Trees	100	100	50	96.0 %
	100	100	100	96.7 %
	100	100	300	97.2 %
	100	100	500	97.1 %
	100	100	750	97.2 %

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 V. The results of changing the parameters are shown in Table IV. The feature used is the HOG 2 descriptor. The performance of the Random Forests does not vary significantly (0 to 2%) when the parameters are changed, which makes them more generic and easier to use.

TABLE V
CLASSIFICATION RESULTS OF THE RANDOM FORESTS USING DIFFERENT HOG DESCRIPTORS

Descriptor	Classification Rate
HOG 1	95.1 %
HOG 2	97.2 %
HOG 3	95.2 %
HOG 4	94.1 %

Table V shows the classification results of the Random Forests for the four different HOG descriptors. 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 20% higher classification rates than the K-d trees when using the HOG 1, 2 and 3 descriptors, but only 1.2% when using HOG 4. Since a small subset 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 all the HOG descriptor values.

Table VI shows the results obtained from building Random Forests using the Distance Transforms. The number of random trees in the forest is set to 500 and 100 variables and samples were chosen at random for their construction. The Random Forests perform better than the K-d trees using the Distance Transforms. The classification rates are increased by up to 30%.

TABLE VI
CLASSIFICATION RESULTS OF VARIOUS FEATURES USING RANDOM FORESTS

Feature	Bottom hat	Canny
DT 50x50	81.2 %	77.8 %
DT 41x41	81.8 %	77.4 %

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we use K-d trees and Random Forests to classify 43 types of traffic signs. Different size HOG descriptors and Distance Transforms are used to evaluate the performance of the these two tree classifiers on images from the German Traffic Sign Benchmark data set [1]. The results show that the finer spatial binning in the HOG descriptor achieves better results. The K-d tree achieves a classification result of 92.9% with HOG descriptors and 67%

with the Distance Transforms. The Random Forests improve the results to 97.2% and 81.8% respectively.

The Random Forests are less sensitive to variations in the background than the K-d trees because of the random selection of the variables. This is shown by testing HOG descriptors which are computed on images with and without a 10% border around the traffic sign. The performance of the K-d tree increases by up to 20% when excluding the border while that of the Random Forests is not influenced by the variations of the background. The advantage of K-d trees over the Random Forests is that they are faster to build, query and update. They also require less memory, since they consist of a single tree rather than an ensemble.

Future work includes using other features such as color HOG and Color Distance Transforms to improve the classification results. In the scope of our previous work on Driver Assistance Systems (DAS), we use an SVM classifier, trained on HOG desriptors, to detect traffic signs in urban and highway environments with varying illumination and weather conditions. The resulting detections are classified by a K-d tree, yielding a classification rate of about 80% at a processing rate of 10 frames per second using a Dual Core CPU 2.4 Ghz.

References

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *submitted to International Joint Conference on Neural Networks*, 2011.

[2] A. De La Escalera, L. Moreno, M. Salichs, and J. Armingol, "Road traffic sign detection and classification," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 848–859, 1997.

[3] 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*. Citeseer, 2005, pp. 255–260.

[4] X. 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*, vol. 17, no. 4, pp. 675–685, 2006.

[5] W.-J. Kuo and C.-C. Lin, "Two-stage road sign detection and recognition," in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 1427 –1430.

[6] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler, "Autonomous driving approaches downtown," *IEEE Intelligent Systems*, vol. 13, no. 6, pp. 1–14, 1999.

[7] A. Vázquez-Reina, S. Lafuente-Arroyo, P. Siegmann, S. Maldonado-Bascón, and F. Acevedo-Rodríguez, "Traffic sign shape classification based on correlation techniques," in *Proceedings of the 5th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*. World Scientific and Engineering Academy and Society (WSEAS), 2005, pp. 149–154.

[8] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognition*, vol. 43, no. 1, pp. 416–430, 2010.

[9] 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*, vol. 8, no. 2, pp. 264–278, 2007.

[10] 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*. IEEE, 2005, pp. 607–612.

[11] B. Alefs, G. Eschemann, H. Ramoser, and C. Beleznai, "Road sign detection from edge orientation histograms," in *2007 IEEE Intelligent Vehicles Symposium*, June 2007, pp. 993–998.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, pp. 886–893. [Online]. Available: http://lear.inrialpes.fr/pubs/2005/DT05

[13] Y. Xie, L. Liu, C. Li, and 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, and 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, and 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] J. Beis and 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.

[17] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," 2007.

[18] T. 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*, vol. 2, 2007, pp. 310 –317.

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.