An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine

Zhiyong Huang, Yuanlong Yu, Jason Gu, and Huaping Liu

***Abstract*—This paper proposes a computationally efficient method for traffic sign recognition (TSR). This proposed method consists of two modules: 1) extraction of histogram of oriented gradient variant (HOGv) feature and 2) a single classifier trained by extreme learning machine (ELM) algorithm. The presented HOGv feature keeps a good balance between redundancy and local details such that it can represent distinctive shapes better. The classifier is a single-hidden-layer feedforward network. Based on ELM algorithm, the connection between input and hidden layers realizes the random feature mapping while only the weights between hidden and output layers are trained. As a result, layer-by-layer tuning is not required. Meanwhile, the norm of output weights is included in the cost function. Therefore, the ELM-based classifier can achieve an optimal and generalized solution for multiclass TSR. Furthermore, it can balance the recognition accuracy and computational cost. Three datasets, including the German TSR benchmark dataset, the Belgium traffic sign classification dataset and the revised mapping and assessing the state of traffic infrastructure (revised MASTIF) dataset, are used to evaluate this proposed method. Experimental results have shown that this proposed method obtains not only high recognition accuracy but also extremely high computational efficiency in both training and recognition processes in these three datasets.**

***Index Terms*—Extreme learning machine (ELM), HOG variant (HOGv), traffic sign recognition (TSR).**

I. Introduction

TRAFFIC sign recognition (TSR) technique has great potential for real-world applications, such as driver assistance systems, autonomous vehicles, and mobile robots based on the fact that it can provide the current state of traffic signs on the road.

Feature representation is one of the important factors for TSR. In order to be easily readable and recognized by drivers, traffic signs are always designed in specific shapes and colors such that the symbols and text in the signs are distinctive to background. For example, stop sign is an octagon combining with a strong word "stop." However, it is a tricky thing to design features of signs for some contaminated conditions, such as bad weather, partial occlusions, viewpoint variations, rotations, and physical damages as shown in Fig. 1. Some robust features in terms of shape, e.g., histogram of oriented gradients (HOG) [1], are widely used for TSR [2]–[6]. In order to represent more local detail information, each cell in the HOG feature is normalized over each of its neighboring blocks, respectively. This can lead to a feature representation with much more dimensions, but this representation is redundant such that it would decay the subsequent classification performance. Thus, how to keep a good balance between redundancy and local details is a challenging issue for designing feature representations of traffic signs.

Fig. 1. Example signs under contaminated conditions including bad weather, partial occlusions, viewpoints variations, rotations, and physical damages.

Furthermore, there are a variety of traffic signs in the real world, e.g., 43 classes are included in the German TSR benchmark (GTSRB) dataset [7]. Therefore, multiclass recognition is another important factor for TSR. Earlier work employed ensemble classifiers, such as one-to-all strategy with support vector machines (SVMs) as base classifiers [4], [8], [9] and random forests with K-d trees as weak classifiers [3]. However, due to their underlying mechanism of binary classification, these methods have to face the unbalance between the number of positive and negative training samples. As a result, these methods are likely to achieve a local optimum or an over-fitting solution. Thus, how to design a classifier that can obtain an optimal and generalized solution for multiclass TSR is the second challenging issue.

Recently, deep neural networks (DNNs), e.g., convolutional neural networks (CNNs) [10]–[12], have been used to automatically learn feature representations of traffic signs. These DNN algorithms combine feature extraction and classification into a unified neural network. They have shown higher recognition accuracy. However, the features learning mechanism in DNNs cannot guarantee robustness to contaminated conditions, e.g., rotation and scaling, unless the training samples can cover various observation conditions as

Manuscript received January 23, 2016; revised February 8, 2016; accepted February 18, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61473089. This paper was recommended by Associate Editor G.-B. Huang. *(Corresponding author: Yuanlong Yu.)*

Z. Huang and Y. Yu are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China (e-mail: hzy_fzusj@sina.com; yu.yuanlong@fzu.edu.cn).

J. Gu is with the Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3H 4R2, Canada (e-mail: jgu@dal.ca).

H. Liu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: hpliu@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2533424

enough as possible. Furthermore, their computational cost during both training and recognition processes is expensive. Due to the high speed of vehicles, not only accuracy but also computational speed should be concerned for real-time applications of TSR. Therefore, no matter whether manually designed features or automatically learned features are used, how to improve computational efficiency while keeping high accuracy and robustness is the third challenging issue for TSR.

This paper proposes an efficient method for TSR. It first extracts variant HOG features of traffic signs and then uses only one extreme learning machine (ELM) [13] for multiclass identification. This proposed method aims to achieve a good balance between recognition accuracy and computational speed.

The HOG variant (HOGv) proposed by this paper has two improvements compared with the original HOG descriptor [1]. First, both contrast sensitive (i.e., signed) and contrast insensitive (i.e., unsigned) orientations of gradients are included such that more detailed local information of signs can be involved into the accumulated histograms. Second, after each cell's oriented histogram is normalized over four of its neighboring blocks, respectively, these normalized histograms of this cell are dimensionally reduced based on a principle component analysis (PCA) like strategy [14] so as to remove redundant information. Therefore, the HOGv feature can address the aforementioned first issue.

ELM [13] is a learning algorithm for single-hidden-layer feedforward neural networks (SFNNs). The first advantage of ELM algorithm is that the input weights between input and hidden layers are randomly assigned. That is, the connection between input and hidden layers realizes a random feature mapping. Since only the output weights between hidden and output layers are trained, layer-by-layer back-propagated tuning is not required. The second advantage is the improved generalization in that the norm of output weights is included in the cost function. Based on these two advantages, ELM algorithm can obtain an optimal and generalized solution for multiclass recognition. Additionally, it is easy to extend ELM to a multilayer network [15] or stacked deep network by using autoencoder technique [16]. ELM has also been used to model local receptive fields [17] and used for representational learning for big data [18]. Therefore, the use of ELM for TSR can give a better solution for the aforementioned second challenging issue.

Furthermore, due to the random assignment of input weights, ELM algorithm can decrease the computational cost of training. As there is only one hidden layer, the computational speed of recognition process is also fast. Thus, as for the aforementioned third issue, the combination of ELM algorithm and HOGv feature can obtain a good balance between recognition accuracy and computational efficiency.

The remainder of this paper is organized as follows. Section II reviews related work on TSR. Section III introduces the framework of this proposed method. Sections IV and V present details about extraction of HOGv feature and ELM-based classification, respectively. Section VI shows experimental results.

II. Related Work

Many studies on TSR have been reported in the past decade. Basically, a TSR system consists of three modules: 1) data preprocessing; 2) feature extraction; and 3) classification.

Data preprocessing is a very useful process to improve feature robustness and recognition accuracy. Thus, various preprocessing methods have been proposed [2], [8], [10]–[12], [19], [20]. In order to cope with illumination changes and high contrast variations, some methods normalize the input images in RGB color space [10]–[12], [19] or in gray space [20], while others convert input images from RGB color space into HSV color space [8]. Recently, some types of transformations, e.g., translation, rotation, and scaling, are made on training images [2], [10]–[12] so as to improve robustness in feature extraction and recognition in the sense that these transformed training images can cover more observation conditions.

Many manually designed features have been proposed for TSR. Since traffic signs are salient in terms of shape, features based on statistics in terms of gradient or orientation energy are widely used to represent traffic signs, e.g., HOG [1]–[6], scale-invariant feature transform (SIFT) [21]–[23], and Gabor features [24], [25]. HOG and SIFT features are both extracted by accumulating oriented gradients except that HOG conducts accumulation around each block while SIFT conducts accumulation around each keypoint. Due to the use of oriented gradients as feature primitives, HOG and SIFT descriptors are both robust to illumination changes. Since HOG descriptor of each cell is normalized over several of its neighbors, the descriptor includes more neighboring information such that it is more discriminative. Furthermore, SIFT feature is robust to rotation and scaling since gradients used for accumulation are relative to the principle orientation and keypoints are extracted based on normalized derivatives (e.g., difference of Gaussian). However, keypoint-based SIFT feature is sparse, and therefore, it has to face the problem of inconsistence of dimensionality due to the difference of keypoint numbers between images. Additionally, features based on statistics in terms of gray level, e.g., local binary patterns [26], are also used to describe texture of traffic signs [27] and have shown great discrimination. In order to complement each other, the combination of some different features is also proposed for TSR [6], [20], [28]. However, combination would lead to a feature representation with much more dimensions. So some techniques [5], [6] have been designed to reduce dimensionality. Recently, some methods [28] quantize the above basic local features using coding techniques, e.g., locality-constrained linear coding [29], and then concatenate these coded features into a global feature representation over the whole image using pooling techniques, e.g., spatial pyramid matching [30].

As for classification, the one-to-all strategy with binary SVMs as the base classifiers [4], [8], [9] is widely used for TSR. Other multiclass identification techniques are also used, e.g., back-propagation neural network (BP-NN) [31] and K-d tree [3]. BP-NN is computational expensive for training and easy to fall into a local optimum. K-d tree has shown comparable performance with other state-of-the-art methods in terms of computational speed of recognition process, but its recognition accuracy is not very high. Random forests are further used

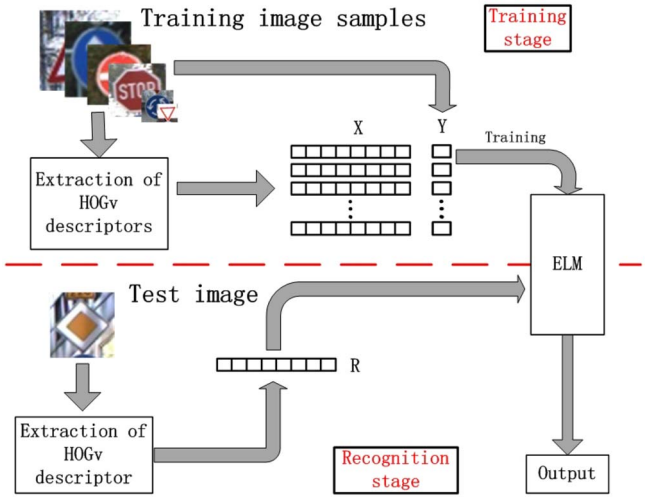


Fig. 2. Framework of this proposed TSR method.

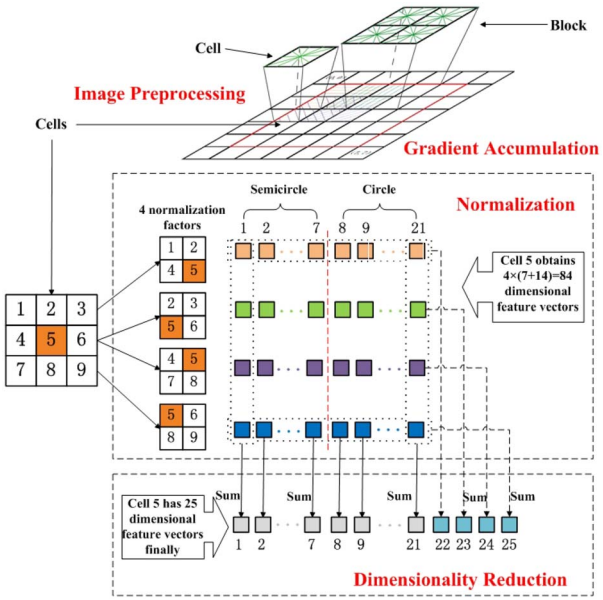


Fig. 3. Extraction of HOGv descriptor.

for TSR [3]. This method achieves higher accuracy, however, its computational cost increases. A two-level hierarchy classification system based on SVMs is also proposed for TSR [2]. This method significantly improves the recognition accuracy but the recognition time increases. ELM has also been used for TSR in our pervious work [32]. Compared with our previous work, this paper has two improvements: 1) proposing an HOGv feature and 2) further using kernel ELM as the classifier.

As a representative of DNN, CNN has been used for TSR and shown impressive recognition accuracy recently [10]–[12]. The feature extraction and classification are combined into a multilayer neural network such that features are learned from input images rather than manually encoded. In order to improve recognition accuracy, an ensemble classifier consisting of 20 CNNs is further proposed [10], [12]. However, these DNN-based methods have a large number of tuning parameters with the result that the computational cost of training is extremely high. Meanwhile, due to their multihidden-layer structure, the computational cost of recognition is also high. Recently, a method [33] has been proposed using CNN to learn features and then using ELM as classifier. This method can obtain competitive results with less computation time compared with CNN methods.

III. Framework of This Proposed Method

Fig. 2 illustrates the framework of this proposed TSR method. This method includes two successive modules: 1) feature extraction module and 2) ELM module. Each training and test image is an instance of traffic signs. In the feature extraction module, an HOGv feature vector is extracted given an input image. Details of HOGv feature extraction can be seen in Section IV. ELM module is a traffic sign classifier composed of an SFNN.

This method consists of two stages: 1) training and 2) recognition. The training stage uses ELM algorithm to estimate output weights of the SFNN given all training images in a batch learning mode. ELM algorithm admits two types of operations in the hidden nodes: 1) dot product and 2) kernel operation. Thus, the corresponding training algorithms are, respectively, denoted as ELM and kernel ELM in this paper. The ELM randomly assigns the input weights while kernel ELM randomly assigns a set of kernels. The training inputs include a feature matrix $\mathbf{X}$ where each row is the feature vector of a training image and a class label vector $\mathbf{Y}$ where each entry indicates which sign class the training image belongs to. $\mathbf{X}$ and $\mathbf{Y}$ are then fed to the ELM module to train the SFNN to make it encapsulate 43 classes of signs. Details of the training process can be seen in Section V.

In the recognition stage, the trained ELM module outputs the class label given the HOGv feature $\mathbf{R}$ of a test image.

IV. Extraction of HOGv Descriptor

As shown in Fig. 3, extraction of HOGv descriptor includes five successive steps: 1) image preprocessing; 2) gradient accumulation; 3) normalization; 4) dimensionality reduction; and 5) concatenation.

A. Image Preprocessing

The input image is preprocessed using the following operations.

1) Scale the image to the fixed size of $w \times h$ pixels using bilinear interpolation.
2) Convert the scaled RGB image into a gray one.
3) Apply gamma correction on the gray image.

B. Gradient Accumulation

Given the preprocessed image with the size of $w \times h$, it is first divided into nonoverlapping cells and each cell is with $k \times k$ pixels. So the index of a cell can be denoted as (p, q) where $0 \leq p < w/k$ and $0 \leq q < h/k$. For each cell, two histograms are accumulated in terms of gradient orientation.

Each pixel in the cell is voted into the corresponding histogram. One histogram is denoted as $\mathbf{C}(p, q)$ and it consists of seven orientation bins over 0°–180° (i.e., covering semicircle orientations). The other histogram is denoted as $\mathbf{D}(p, q)$ and it consists of 14 orientation bins over 0°–360° (i.e., covering circle orientations). It can be seen that both contrast sensitive and insensitive gradient orientations are included. Thus, more detailed local information of signs can be involved in these histograms.

C. Normalization

In order to involve more neighboring information, 2×2 cells are grouped into a block. These blocks are overlapping and their spacing stride is k pixels. For each cell, either histogram is normalized using the following two routines.

1) Estimate a gradient energy measure for a block containing this cell.
2) Use this measure to normalize the histogram of this cell.

Since each cell belongs to four neighboring blocks except the cells in the image's margin, it is normalized using the above routines four times. For histograms $\mathbf{C}(p, q)$ and $\mathbf{D}(p, q)$ of the cell indexed by (p, q), four energy measures, respectively, denoted as $NC_{\delta,\gamma}(p, q)$ and $ND_{\delta,\gamma}(p, q)$ where $\delta, \gamma \in \{-1, 1\}$, are estimated as follows:

$$NC_{\delta,\gamma}(p,q) = \Big[\|\mathbf{C}(p,q)\|^2 + \|\mathbf{C}(p+\delta,q)\|^2 + \|\mathbf{C}(p,q+\gamma)\|^2 + \|\mathbf{C}(p+\delta,q+\gamma)\|^2\Big]^{\frac{1}{2}} \tag{1}$$

and

$$ND_{\delta,\gamma}(p,q) = \Big[\|\mathbf{D}(p,q)\|^2 + \|\mathbf{D}(p+\delta,q)\|^2 + \|\mathbf{D}(p,q+\gamma)\|^2 + \|\mathbf{D}(p+\delta,q+\gamma)\|^2\Big]^{\frac{1}{2}}. \tag{2}$$

Thus, histograms $\mathbf{C}(p, q)$ and $\mathbf{D}(p, q)$ of the cell (p, q) can be normalized using (3) to obtain a feature representation $\mathbf{F}(p, q)$ (in a matrix format) with a total of $4 \times (7 + 14) = 84$ dimensions.

It is important to note that the normalization operation is ignored for the cells in the image margin and it is only applied into the cells within the red box as shown in Fig. 3

$$\mathbf{F}(p,q) = \begin{pmatrix} \mathbf{C}(p,q)/NC_{-1,-1}(p,q), \mathbf{D}(p,q)/ND_{-1,-1}(p,q) \\ \mathbf{C}(p,q)/NC_{+1,-1}(p,q), \mathbf{D}(p,q)/ND_{+1,-1}(p,q) \\ \mathbf{C}(p,q)/NC_{+1,+1}(p,q), \mathbf{D}(p,q)/ND_{+1,+1}(p,q) \\ \mathbf{C}(p,q)/NC_{-1,+1}(p,q), \mathbf{D}(p,q)/ND_{-1,+1}(p,q) \end{pmatrix}. \tag{3}$$

D. Dimensionality Reduction

A PCA-like strategy [14] is used to reduce the redundant information of each 84-dimensional feature representation. This reduction operator consists of 21 column summations and four row summations. A column summation captures the overall gradient energy of the corresponding orientation over spatial neighbors. Since seven orientation bins over 0°–180° and 14 orientation bins over 0°–360° are included in a feature representation $\mathbf{F}(p, q)$, there are a total of 21 column summations. A row summation captures the overall gradient energy of a block containing the cell (p, q) over orientations. Since each cell (p, q) has four neighboring blocks, there are totally four row summations. Therefore, the final cell-based feature vector $\mathbf{F}'(p, q)$ has 25 dimensions.

E. Concatenation

The cell-based features of 2×2 cells within each block are stacked together to form a higher dimensional feature vector for each block, called block-based feature vector. Since blocks are spatially overlapped, the block-based feature vectors can jointly encode more neighborhood information [34]. All block-based feature vectors in the image are concatenated to finally form an HOGv descriptor $\mathbf{x}$.

V. ELM-Based Training and Recognition

A. Structure of ELM-Based Classifier

ELM [13] is basically a machine learning algorithm for training the SFNN.

The input layer is connected to the input feature vector $\mathbf{x}$ (i.e., HOGv descriptor) of an image of traffic sign. The dimension number of $\mathbf{x}$ is denoted as P.

At the hidden layer, the number of hidden nodes is denoted as L. The output of a hidden node indexed by i is denoted as $g(\mathbf{x}; \mathbf{w}_i, b_i) = g(\mathbf{x}\cdot\mathbf{w}_i + b_i)$, where g is the activation function, $\mathbf{w}_i$ is the input weight vector between this hidden node and all input nodes, b_i is the bias of this node and $i = 1, \ldots, L$. The connection between input and hidden layers is actually a function of feature mapping from a P-dimensional space to an L-dimensional space. Given an input feature $\mathbf{x}$, its mapped feature vector can be denoted as

$$\mathbf{h}(\mathbf{x}) = \big[g(\mathbf{x}; \mathbf{w}_1, b_1), \ldots, g(\mathbf{x}; \mathbf{w}_L, b_L)\big]. \tag{4}$$

It has been proved that the universal approximation can be satisfied if the activation function g is a nonlinear piecewise continuous function [35]–[37]. In this paper, sigmoid function as shown in (5) is used as the activation function

$$g(\mathbf{x}; \mathbf{w}_i, b_i) = \frac{1}{1 + \exp[-(\mathbf{x}\cdot\mathbf{w}_i + b_i)]}. \tag{5}$$

At the output layer, the number of output nodes is denoted as M. M is equal to the number of traffic sign classes, i.e., each output node represents a traffic sign class. The output weight between the ith hidden node and the jth output node is denoted as $\beta_{i,j}$, where $j = 1, \ldots, M$. The value of an output node j can be calculated as

$$f_j(\mathbf{x}) = \sum_{i=1}^{L} \beta_{i,j} \times g(\mathbf{x}; \mathbf{w}_i, b_i). \tag{6}$$

Thus, for the input sample $\mathbf{x}$, its output vector at the hidden layer can be written as

$$\mathbf{f}(\mathbf{x}) = \big[f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})\big] = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{7}$$

Algorithm 1 Training Routine of an ELM-Based Classifier

1: Given a set of training samples $\{(\mathbf{x}_k, \mathbf{y}_k)_{k=1,\ldots,N}\}$, activation function g, and hidden node number L:
2: Step 1: Randomly generate hidden neuron parameter (e.g., input weight vector $\mathbf{w}_i$ and bias b_i), where $i = 1, \ldots, L$, based on some continuous distribution (e.g., uniform distribution);
3: Step 2: Calculate matrix $\mathbf{H}$;
4: Step 3: Estimate the output weight $\boldsymbol{\beta}$ based on some optimization constraints (e.g., (14) or (15)).

where

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_L \end{bmatrix} = \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{1,M} \\ \vdots & \vdots & \vdots \\ \beta_{L,1} & \cdots & \beta_{L,M} \end{bmatrix}. \tag{8}$$

During the recognition process, given a test sample $\mathbf{x}$, its class label of $\mathbf{x}$ can be determined as

$$\text{label}(\mathbf{x}) = \arg_{j=1,\ldots,M} \max f_j(\mathbf{x}). \tag{9}$$

B. Training Process of ELM-Based Classifier

The supervised training requires N training sample pairs, each of which consists of a feature vector $\mathbf{x}_k$ and its binary class label vector (i.e., ground truth) $\mathbf{t}_k = [t_{k,1}, \ldots, t_{k,M}]$, where $k = 1, \ldots, N$. In the label vector, each entry indicates whether or not the sample $\mathbf{x}_k$ belongs to the corresponding class. All labels can form a matrix denoted as $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$.

It can be seen that the training parameters for an ELM include two parts: 1) the input weights and biases $\{\mathbf{w}_i, b_i\}_{i=1,\ldots,L}$ and 2) the output weight matrix $\boldsymbol{\beta}$ as shown in (8). In the ELM algorithm, the input weights and biases are randomly assigned. Therefore, only $\boldsymbol{\beta}$ is trained.

Let $\mathbf{y}_k$ denote the actual output vector for the input $\mathbf{x}_k$. Taking all training samples $\{\mathbf{x}_k\}$ into (6) can form a linear representation

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \tag{10}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} g(\mathbf{x}_1; \mathbf{w}_1, b_1) & \cdots & g(\mathbf{x}_1; \mathbf{w}_L, b_L) \\ \vdots & \vdots & \vdots \\ g(\mathbf{x}_N; \mathbf{w}_1, b_1) & \cdots & g(\mathbf{x}_N; \mathbf{w}_L, b_L) \end{bmatrix} \tag{11}$$

and

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,M} \\ \vdots & \ddots & \vdots \\ y_{N,1} & \cdots & y_{N,M} \end{bmatrix}. \tag{12}$$

The training process aims to minimize the training error $\|\mathbf{T} - \mathbf{H}\boldsymbol{\beta}\|^2$ and the norm of output weight $\|\boldsymbol{\beta}\|$ [38]. So the training process can be represented as a constrained-optimization problem

$$\begin{aligned} \text{minimize:} \quad & \Psi(\boldsymbol{\beta}, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\boldsymbol{\xi}\|^2 \\ \text{subject to:} \quad & \mathbf{H}\boldsymbol{\beta} = \mathbf{T} - \boldsymbol{\xi} \end{aligned} \tag{13}$$

where constant C is used as a regularization factor to control the tradeoff between the closeness to the training data and the smoothness of the decision function such that generalization performance is improved.

Lagrange multiplier technique is used to solve the above optimization problem [38]. If matrix $((\mathbf{I}/C) + \mathbf{H}^T\mathbf{H})$ is not singular, solution $\boldsymbol{\beta}$ can be obtained as

$$\boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{T}. \tag{14}$$

If matrix $((\mathbf{I}/C) + \mathbf{H}\mathbf{H}^T)$ is not singular, solution $\boldsymbol{\beta}$ can be obtained as

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}. \tag{15}$$

It can be seen that the dimensionality of $((\mathbf{I}/C) + \mathbf{H}^T\mathbf{H})$ is $L \times L$ while $((\mathbf{I}/C) + \mathbf{H}\mathbf{H}^T)$ is $N \times N$. Therefore, if the number of training samples is huge, the solution in (14) can be used to decrease computational cost; otherwise, the solution in (15) can be used.

The training routine for the ELM-based classifier used in this paper is shown in Algorithm 1.

Compared with other learning algorithms, e.g., BP algorithm, for neural networks, ELM randomly sets input weights and biases at the hidden layer without training such that the output weights can be quickly estimated. It can be seen that there are only two tuning parameters: one is the number of hidden nodes (i.e., L) and the other is the regularization factor (i.e., C).

C. Kernel ELM-Based Classifier

In the case that feature mapping function $\mathbf{h}(\mathbf{x})$ is unknown, kernel technique can be applied into ELM based on Mercer's condition. That is, given two input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the dot product of their mapped features $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j)$ can be replaced by a kernel function $\phi(\mathbf{x}_i, \mathbf{x}_j)$.

Therefore, based on (15), the output vector $\mathbf{f}(\mathbf{x})$ of a kernel ELM can be represented as

$$\begin{aligned} \mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} &= \mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \\ &= \begin{bmatrix} \phi(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ \phi(\mathbf{x}, \mathbf{x}_{N_k}) \end{bmatrix}\left(\frac{\mathbf{I}}{C} + \boldsymbol{\Phi}\right)^{-1}\mathbf{T} \end{aligned} \tag{16}$$

where

$$\boldsymbol{\Phi} = \mathbf{H}\mathbf{H}^T = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1, \mathbf{x}_{N_k}) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_{N_k}, \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_{N_k}, \mathbf{x}_{N_k}) \end{bmatrix} \tag{17}$$

TABLE I
PARAMETERS OF DIFFERENT HOG DESCRIPTORS

Name	Image Size	Block Size	Cell Size	Stride	Gradients	Bins	Dimensions	10% Margin	Rotated Training Images
HOGs_b	48×48	12×12	6×6	6×6	Semicircle	7	1372	Remained	None
HOGs_nb	48×48	12×12	6×6	6×6	Semicircle	7	1372	Removed	None
HOGc_b	48×48	12×12	6×6	6×6	Circle	7	1372	Remained	None
HOGc_nb	48×48	12×12	6×6	6×6	Circle	7	1372	Removed	None
HOGv_b	48×48	12×12	6×6	6×6	Semicircle Circle	7 14	2500	Remained	None
HOGv_nb	48×48	12×12	6×6	6×6	Semicircle Circle	7 14	2500	Removed	None
HOGv+r	48×48	12×12	6×6	6×6	Semicircle Circle	7 14	2500	Removed	Included

and N_k denotes the number of training samples used for the kernel ELM. These N_k samples are randomly selected from the training set.

In this paper, Gaussian function is used as the kernel ϕ

$$\phi(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \tag{18}$$

where σ denotes the spread (i.e., standard deviation) of the Gaussian function.

VI. EXPERIMENTS

A. Datasets

Three benchmark datasets are used to validate the proposed method: 1) the GTSRB dataset[1] [7]; 2) the Belgium traffic sign classification (BTSC) dataset[2] [39]; and 3) the revised mapping and assessing the state of traffic infrastructure (revised MASTIF) dataset [40].[3]

GTSRB dataset covers 43 classes of traffic signs. In this dataset, training set includes 39 209 training images while test set includes 12 630 test images. These images vary in size from 15×15 to 250×250 pixels and some of them are not square.

BTSC dataset, as a subset of Belgium traffic sign dataset [39], was built for evaluating TSR. This dataset covers 62 classes of traffic signs with 4591 images in the training set and 2534 images in the test set.

Revised MASTIF dataset, derived from MASTIF [41] dataset, was also built for evaluating TSR. This dataset covers 31 classes of traffic signs and there are 4044 training images and 1784 test images.

B. Experimental Setup

1) Data Preprocessing: In our experiments, all images are rescaled to the same size of 48×48 pixels using bilinear interpolation before feature extraction.

2) Setup for Evaluating Different HOG Descriptors: In order to evaluate the performance of this proposed HOGv descriptor, this paper uses another two types of HOG descriptors, denoted as HOGs and HOGc, respectively, as competing descriptors. HOGs descriptor is obtained based on semicircle gradient orientations while HOGc descriptor is obtained based on circle gradient orientations.

In order to evaluate the influence of background on HOG descriptors, each image in the datasets has a background-removed version which is obtained by removing around 10% margin using a tight bounding box. Therefore, the above three descriptors, including HOGv, HOGs, and HOGc, are extracted for each image and its background-removed version, respectively. That is, each image has six descriptors, denoted as HOGv_b, HOGv_nb, HOGs_b, HOGs_nb, HOGc_b, and HOGc_nb. More details can be seen in Table I.

Furthermore, in order to evaluate the robustness to rotation, rotated versions (within the range of $\pm 15°$) of some training images are added into the training set in our experiments. HOGv_nb descriptor is extracted for each image in this extended training set and such descriptor is denoted as HOGv+r.

Extracting HOGs, HOGc, and HOGv is completed by our own developed codes. Implementations of ELM and kernel ELM algorithms are based on their online MATLAB codes.[4]

Two points should be noted for HOG evaluation.

1) In order to give a statistical evaluation, ten training-test data partitions are randomly selected in each dataset.
2) These HOG descriptors are evaluated under the same data preprocessing, same training-test data partition, same classifier, and same computation platform.

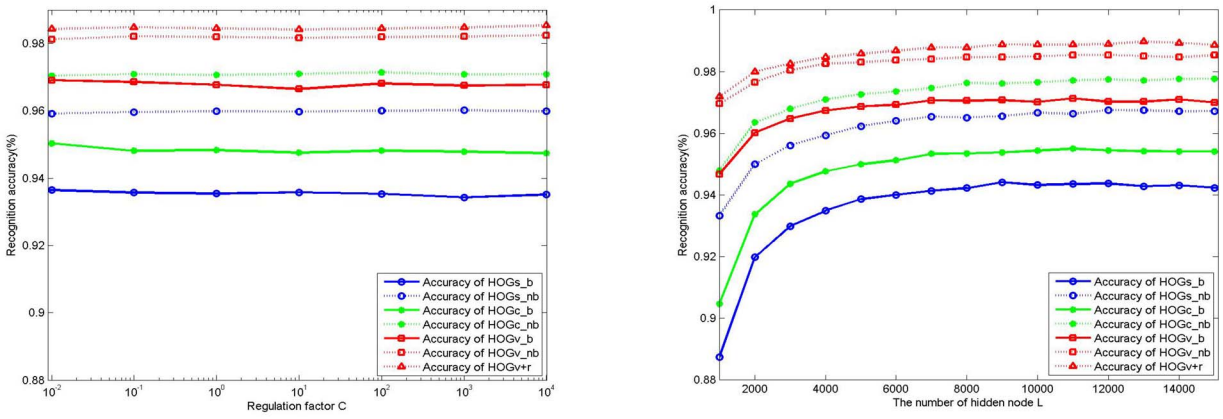
3) Setup for Evaluating Competing Classifiers: In order to evaluate the performance of ELM and kernel ELM, this paper uses SVM, kernel SVM (with Gaussian kernel), and linear discriminant analysis (LDA) as competing classifiers. Recognition accuracy and computational efficiency (i.e., training time and recognition time) are used as evaluation measures.

In our experiments, SVM and kernel SVM algorithms are implemented using LIBSVM package[5] [42]. LDA algorithm is implemented using Shark library.[6] One-to-all strategy is used on SVM, kernel SVM, and LDA.

Two points should be noted for classifier evaluation.

1) In order to give a statistical evaluation, ten training-test data partitions are randomly selected in each dataset.
2) All these five types of classifiers are evaluated under the same data preprocessing, same training-test data partition, same HOG descriptor, and same computation platform.

[1] http://benchmark.ini.rub.de/.
[2] http://homes.esat.kuleuven.be/~rtimofte.
[3] http://www.zemris.fer.hr/~kalfa/Datasets/rMASTIF/.
[4] http://www.ntu.edu.sg/home/egbhuang/.
[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.
[6] http://image.diku.dk/shark/.



(a) (b)

Fig. 4. Effects of regularization factor C and number of hidden nodes L in ELM-based traffic sign classifier on GTSRB dataset. Average recognition accuracy curves with respect to (a) parameter C and (b) parameter L.

4) Setup for Overall Performance Comparison: This proposed TSR method is finally compared with other published TSR methods in terms of recognition accuracy and computational efficiency. This paper directly adopts the performance data (e.g., recognition accuracy) published in their corresponding papers for comparison. Although various methodologies, e.g., different feature extraction and classification techniques, are used for these published methods, this type of comparison can give an overall evaluation at the level of integrated system.

Two points should be noted for overall performance comparison.

1) Different TSR methods have been published for each dataset, e.g., there are 11 methods published for GTSRB dataset and two methods published for BTSC dataset, but there is no published method found for revised MASTIF dataset.
2) Published methods use the original training-test partition in the benchmark datasets, so this proposed method also uses the same data partition for the overall performance comparison.

5) Computation Platform: A standard PC is used in our experiments and its hardware configuration is as follows.

1) *CPU:* Intel Xeon i7-4790 (3.60 GHz) $\times 1$.
2) *Memory:* 32 GB DDR3.
3) *Graphics processing unit (GPU):* None.

C. Evaluation on GTSRB Dataset

1) Parameter Tuning of ELM Classifier: There are two tuning parameters for the ELM-based traffic sign classifier: 1) regularization factor C and 2) number of hidden nodes (i.e., L). Recognition accuracy is used as a basic performance measure with respect to these tuning parameters.

Fig. 4 shows the average recognition accuracy curves obtained by ELM-based classifier using different types of HOG descriptors. Fig. 4(a) illustrates the average accuracy curves with respect to C given $L = 4000$. It can be seen that accuracy curves have tiny fluctuation as parameter C varies. It means that regularization factor C imposes little influence on recognition performance. Fig. 4(b) illustrates the accuracy curves with respect to L given $C = 2000$. It can be seen that recognition accuracy goes up obviously with the increase of L. However, this trend slows down when L is up to 8000 and accuracy becomes stable when L further increases. It indicates that the ELM-based classifier is not very sensitive to the number of hidden nodes as long as it is set large enough (e.g., $L > 8000$). Thus, this paper sets $C = 2000$ and $L = 10\,000$ in the ELM-based classifier.

The kernel ELM-based classifier also has two tuning parameters, including Gaussian kernel spread σ and regularization factor C.

Fig. 5 shows the recognition accuracy curves obtained by kernel ELM-based classifier using different types of HOG descriptors. Fig. 5(a) illustrates the accuracy curves with respect to σ given $C = 150$. It can be seen that recognition accuracy goes up with the increase of σ until it reaches a peak value and then it goes down with the increase of σ. It indicates that the spread of kernel should be within a medium range. Fig. 5(b) illustrates the accuracy curves with respect to C given a suitable value of σ. It can be seen from Fig. 5(b) that recognition accuracy goes up with the increase of C and then it keeps nearly constant with the increase of C. Therefore, this paper sets $\sigma = 75$ and $C = 250$ for kernel ELM-based classifier.

It can be generally concluded that recognition accuracy is not very sensitive to these tuning parameters no matter whether ELM or kernel ELM-based classifier is used.

2) Evaluation of HOG Descriptors: Given the same classifier, the influence of background on HOG descriptors is first evaluated. Given the same ELM classifier, Fig. 4 illustrates the average recognition accuracy obtained by three types of HOG descriptors (i.e., HOGs_b, HOGc_b, and HOGv_b) and their background removed versions (i.e., HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r). Analogously, given the same kernel ELM classifier, Fig. 5 illustrates the average recognition accuracy obtained by three types of HOG descriptors and their background removed versions. From these two figures, one conclusion can be obtained.

1) The background removed versions (i.e., removing 10% image margin) can have 1% increase in terms of recognition accuracy compared with their corresponding original version.

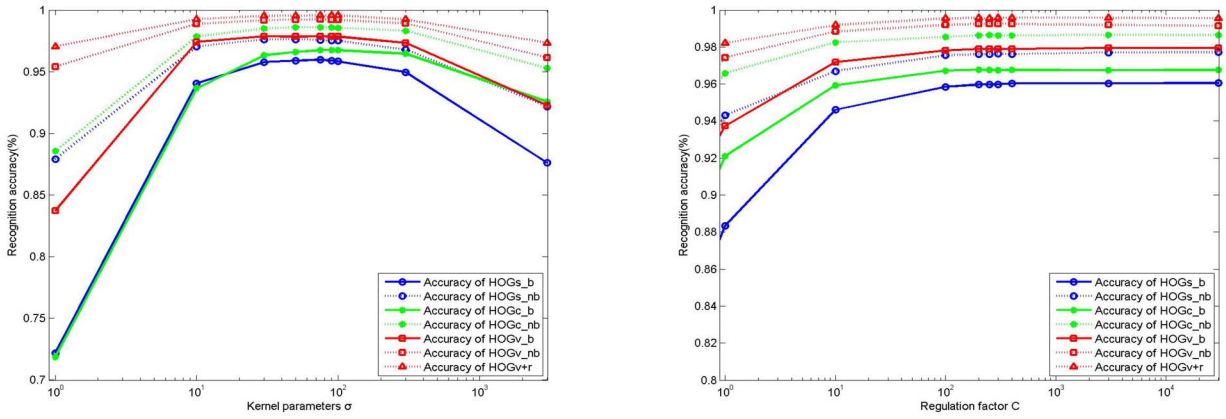


Fig. 5. Effects of Gaussian kernel spread σ and regularization factor C in kernel ELM-based traffic sign classifier on GTSRB dataset. Average recognition accuracy curves with respect to (a) parameter C and (b) parameter σ.

TABLE II
AVERAGE RECOGNITION ACCURACY OBTAINED BY DIFFERENT HOG DESCRIPTORS AND COMPETING CLASSIFIERS ON GTSRB DATASET

	HOGs_nb	HOGc_nb	**HOGv_nb**	**HOGv+r**
Kernel ELM based	97.52±0.30%	98.44±0.16%	98.99±0.10%	**99.49±0.05%**
Kernel SVM based	96.96±0.30%	98.01±0.25%	98.41±0.25%	**98.95±0.15%**
ELM based	96.75±0.16%	97.77±0.11%	98.54±0.12%	**98.89±0.10%**
SVM based	96.12±0.20%	97.43±0.17%	97.65±0.25%	**97.80±0.21%**
LDA based	94.01±0.15%	96.24±0.25%	97.14±0.21%	**97.56±0.20%**

Then all background removed versions (i.e., HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r) are evaluated given each type of classifiers to show which descriptor is better. Average and maximum of recognition accuracy are used for evaluation as shown in Tables II and III, respectively.

In Table II, each row illustrates the performance comparison among HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r given a classifier. From this table, it can be seen that HOGv+r achieves the highest recognition accuracy for each classifier and HOGv_nb also outperforms HOGs_nb and HOGc_nv. It can be further seen that HOGv+r and HOGv_nb show the above effects not only using ELM classifier but also using other classifiers, such as SVM, kernel SVM, and LDA.

In Table III, the third column also illustrates the performance comparison among HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r given each classifier. The above effects can be also seen from Table III.

Thus, the following two conclusions can be summarized from the above analysis.

1) This proposed HOG descriptor with the combination of signed and unsigned gradients (i.e., HOGv_nb) can outperform the original HOG descriptor only with signed gradients (i.e., HOGc_nb) or unsigned gradients (i.e., HOGs_nb).
2) For this proposed HOG descriptor (i.e., HOGv_nb), adding the rotated versions of some training images to the training set (i.e., HOGv+r) can further improve the recognition accuracy.

3) Evaluation of Competing Classifiers: This proposed ELM and kernel ELM classifiers are compared against other competing classifiers (i.e., SVM, kernel SVM, and LDA) using each type of HOG descriptors including HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r. Average and maximum of recognition accuracy, average training time, and average recognition time are used to evaluate these competing classifiers.

Table II lists the average recognition accuracy. In this table, each column illustrates the performance comparison among competing classifiers given an HOG descriptor. As highlighted by red in Table II, it can be seen that kernel ELM achieves the highest recognition accuracy for each type of HOG descriptors.

Table III lists the maximal recognition accuracy, average training time, and average recognition time obtained by each of competing classifiers given each type of HOG descriptors. It is important to note that all of the time listed in Table III includes the time of HOG extraction. As highlighted by red in Table III, it can be seen that kernel ELM achieves the highest recognition accuracy for each type of HOG descriptors. Furthermore, it can be also seen that the kernel ELM takes least training time among these competing classifiers and kernel ELM takes less recognition time than SVM and kernel SVM.

From these two tables, two points can be concluded.

1) This proposed kernel ELM-based TSR method outperforms SVM, kernel SVM, and LDA-based methods in terms of recognition accuracy and training time.
2) Although kernel ELM-based TSR method takes a little more recognition time than LDA-based method, its recognition accuracy is higher than LDA.

4) Overall Performance Comparison: For GTSRB Dataset, there are totally 11 published methods which can be used for overall performance comparison. Results of these methods are obtained from GTSRB website (marked with "∗" in Table IV)

TABLE III
EVALUATION OF DIFFERENT HOG DESCRIPTORS AND COMPETING CLASSIFIERS ON GTSRB DATASET

Method	Descriptor	Maximal Recognition Rate	Average Training Time	Average Recognition Time
Kernel_ELM based	HOGs_nb	97.86%	117s/dataset	2.38ms/frame
	HOGc_nb	98.65%	118s/dataset	2.38ms/frame
	HOGv_nb	99.12%	128s/dataset	3.33ms/frame
	HOGv+r	**99.56%**	209s/dataset	3.88ms/frame
Kernel_SVM based	HOGs_nb	97.17%	4.1758e+03s/dataset	71.26ms/frame
	HOGc_nb	98.12%	5.5012e+03s/dataset	72.84ms/frame
	HOGv_nb	98.56%	3.8370e+03s/dataset	72.05ms/frame
	HOGv+r	**99.03%**	5.3115e+03s/dataset	73.63ms/frame
ELM based	HOGs_nb	96.95%	276s/dataset	2.30ms/frame
	HOGc_nb	97.91%	286s/dataset	2.38ms/frame
	HOGv_nb	98.62%	290s/dataset	3.01ms/frame
	HOGv+r	**99.09%**	295s/dataset	3.17ms/frame
SVM based	HOGs_nb	96.25%	2.1210e+03s/dataset	37.21ms/frame
	HOGc_nb	97.66%	2.0504e+03 s/dataset	34.05ms/frame
	HOGv_nb	97.89%	1.9922e+03s/dataset	45.13ms/frame
	HOGv+r	**97.94%**	2.5816e+03s/dataset	42.76ms/frame
LDA based	HOGs_nb	94.18%	145s/dataset	0.51ms/frame
	HOGc_nb	96.44%	135s/dataset	0.51ms/frame
	HOGv_nb	97.34%	320s/dataset	0.53ms/frame
	HOGv+r	**97.71%**	370s/dataset	0.53ms/frame

TABLE IV
OVERALL PERFORMANCE COMPARISON OF PUBLISHED METHODS ON GTSRB DATASET

Method	Recognition rate	Training Time	Recognition time	Configuration
HLSGD [12]	99.65%	>7h/dataset	N/A	CPU: I7-3960X GPU: 2 × Tesla C2075
Kernel ELM based	99.56%	209s/dataset	3.9ms/frame	CPU: I7-4790
Hierarchical SVM [2]	99.52%	N/A	40ms/frame	CPU: I3
Committee of CNNs* [10]	99.46%	37h/dataset	11.5ms/frame	CPU: I7-950 GPU: 4 × GTX580
CNN-ELM [33]	99.40%	5h/dataset	N/A	CPU: 8 × E5-2643
ELM based	99.09%	293s/dataset	3.2ms/frame	CPU: I7-4790
Multi-Scale CNNs* [11]	98.84%	N/A	N/A	N/A
INNC+INNLP [6]	98.53%	N/A	47ms/frame	CPU: AMD Opteron 8360SE
SRGE [5]	98.19%	20min/dataset	N/A	CPU: I7-950
LOEMP [27]	97.26%	N/A	70ms/frame	CPU: E7500
Random forests* [3]	96.14%	N/A	N/A	N/A
LDA* [7]	95.68%	N/A	N/A	N/A
K-d trees* [3]	92.70%	N/A	17.9ms/frame for highest accuracy	N/A

or published papers. These published methods are listed as follows.

1) *Committee of CNNs [10]:* It is based on a collection of CNNs for TSR. Features are learned in these CNNs.
2) *Multiscale CNNs [11]:* It is based on a set of multiscale CNNs for TSR. Features are learned in these CNNs.
3) *HLSGD [12]:* It proposes a new method, called hinge loss stochastic gradient descent (HLSGD), to train CNNs. Features are learned in these CNNs.
4) *CNN-ELM [33]:* It uses ELM to build the classifier and uses CNN to learn features.
5) *Hierarchical SVMs [2]:* It is based on a hierarchical classification structure where SVMs are used as base classifiers. It uses HOGc_nb descriptors as features.
6) *INNC+INNLP [6]:* It extracts different features and merges them directly. It then uses iterative nearest neighbors-based linear projections (INNLP) technique to reduce the dimensionality of the merged features and uses iterative nearest neighbors (INNC) technique to build the classifier.
7) *SRGE [5]:* It proposes a combination of sparse representation and graph embedding (SRGE) under LDA framework and uses HOGc_b as basic features.
8) *K-d Trees [3]:* It uses K-d trees as the classifier and uses HOGc_nb as features.
9) *Random Forests [3]:* It uses random forests to build the classifier and uses HOGc_nb as features.
10) *LDA [7]:* It uses LDA technique to build the classifier and uses HOGc_nb as features.
11) *Color Global LOEMP [27]:* It extracts color global and local oriented edge magnitude patterns (LOEMP) as traffic sign features. One-to-all SVMs are used to build the classifier.

a) Overall recognition accuracy: Table IV lists the recognition accuracy of this proposed method and others. The recognition accuracy of this proposed method reaches 99.56% (achieved by kernel ELM-based classifier). This proposed method outperforms some CNN-based methods (e.g., committee of CNNs [10] and multiscale CNNs [11]) and it can rank to the second position among all of these 13 methods in terms of accuracy.

TABLE V
RECOGNITION ACCURACY FOR CATEGORIES ON GTSRB DATASET

	Speed limits	Other prohibitions	Derestriction	Mandatory	Danger	Unique
Kernel ELM based	**99.54%**	**100.00%**	98.33%	**99.94%**	98.96%	**99.95%**
Hierarchical SVM [2]	N/A	N/A	98.89%	**99.94%**	99.03%	99.90%
Committee of CNNs* [10]	99.47%	99.93%	**99.72%**	99.89%	**99.07%**	99.22%
Human best* [7]	98.32%	99.87%	98.89%	100.00%	99.21%	100.00%
ELM based	99.14%	99.80%	96.94%	99.77%	97.81%	99.90%
Human average* [7]	97.63%	99.93%	98.89%	99.72%	98.67%	100.00%
Multi-Scale CNNs* [11]	98.61%	99.93%	98.89%	97.18%	98.67%	98.63%
Random forests* [3]	95.95%	99.13%	87.50%	99.27%	92.08%	98.73%
LDA* [7]	95.37%	96.80%	85.83%	97.18%	93.73%	98.63%

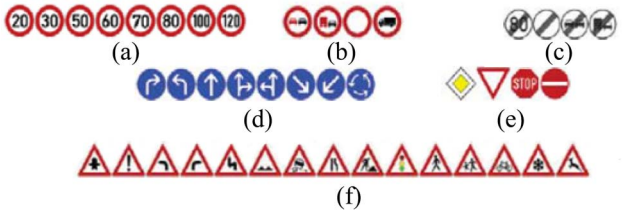


Fig. 6. Categories of traffic signs in GTSRB dataset. (a) Speed limits. (b) Other prohibitions signs. (c) Derestriction signs. (d) Mandatory signs. (e) Unique signs. (f) Danger signs.

b) Overall computational efficiency: Table IV also lists the training time and recognition time of this proposed method and others. It is important to note that both training time and recognition time of this proposed method include the time of HOG extraction. Moreover, since the computation platforms are different, the computational time of this proposed method is only compared with the published methods (e.g., CNN-based methods) whose computation configuration is higher than ours.

As shown in Table IV, although CNN-based HLSGD method [12] is 0.09% higher in terms of accuracy than this proposed method, it is computationally very expensive during both training and recognition. Actually, most CNN-based TSR methods [10], [12] have to face the issue of computational efficiency although they can achieve high accuracy. However, it can be seen from Table IV that this proposed method outperforms these CNN-based methods (see [10], [12]) much in term of computational efficiency. The training process of this proposed method is thousands of times faster than CNN-based methods (see [10], [12]) and the recognition process is also several times faster than CNN-based methods (see [10]). It is important to note that the hardware configuration in our experiments is a standard PC while these CNN-based methods use high-performance GPUs.

c) Accuracy of each category: Since sample numbers between traffic sign classes are unbalanced in GTSRB dataset, evaluation for each category is required. The traffic signs can be split into six categories as shown in Fig. 6, including speed limits, other prohibition signs, derestriction signs, mandatory signs, unique signs, and danger signs. Table V lists each category's recognition accuracy obtained by nine methods. It can be seen that this proposed method achieves the highest accuracy in categories of speed limits, other prohibition signs, mandatory signs, and unique signs. However, committee of CNNs outperforms this proposed method in the category of derestriction signs. This can be explained as follows: the HOGv+r descriptor used in this proposed method is powerful to represent distinctive shapes (e.g., unique signs) while its ability to represent round shape (e.g., derestriction signs) is a little lower. In contrast, convolution operation used in CNNs makes it much easier to encapsulate round shape.

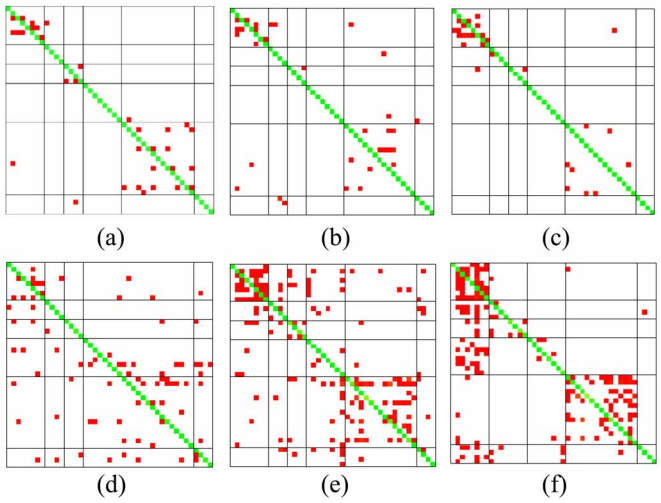
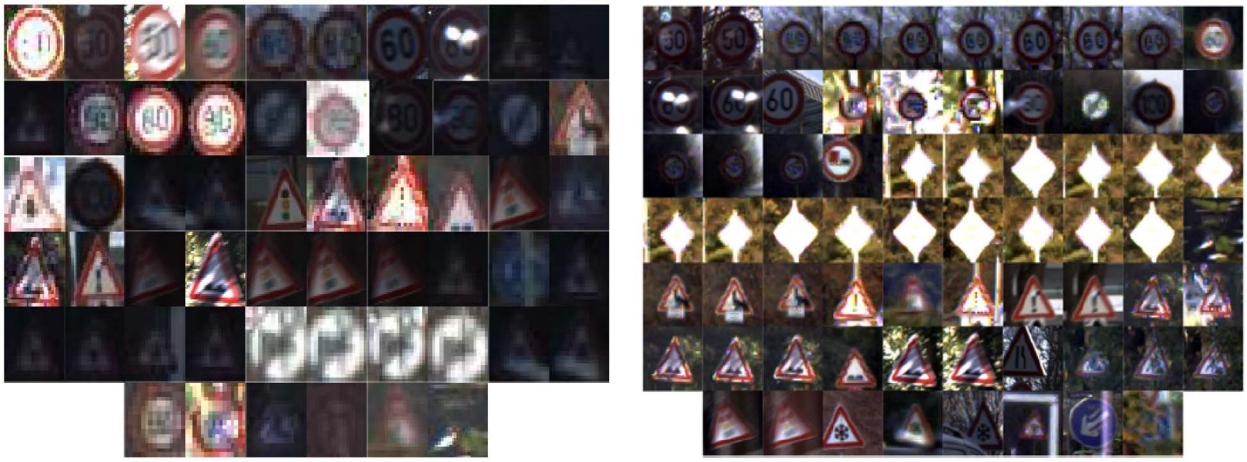


Fig. 7. Patterns of confusion across 43 traffic sign classes on GTSRB dataset. (a) Kernel ELM based. (b) IDSIA-committee of CNNs. (c) Human performance (best individual). (d) Sermanet-multi-scale CNN. (e) CAOR-random forests. (f) LDA.

d) Evaluation in terms of confusion matrix: Detailed evaluation at the class-level is also given in terms of confusion matrix. Fig. 7 shows the patterns of confusion across classes obtained by this proposed method (kernel ELM-based), committee of CNNs, multiscale CNNs, random forests, LDA, and human performance. In this figure, coordinates in x- and y-axis denote 43 traffic sign classes. Red point with coordinates (x, y) represents that there are misclassifications of test samples whose ground truth labels are x while machine's output labels are y. It can be seen that this proposed method shows fewer points in the nondiagonal region (i.e., fewer false positives and false negatives) than methods of multiscale CNNs, random forests and LDA. It indicates that this proposed method outperforms other methods while it shows comparable performance with committee of CNNs.

Fig. 7 also shows the patterns of confusion at the category level. In this figure, categories are separated by grid lines. It can be seen that the committee of CNNs produces some misclassified patterns which are so far away from the grid (i.e., rectangle) where their ground truth category locates. In contrast, most of misclassified patterns produced by this proposed



(a) (b)

Fig. 8. All misclassified images produced by (a) this proposed kernel ELM-based method and (b) committee of CNNs* in GTSRB dataset.

TABLE VI
AVERAGE RECOGNITION ACCURACY OBTAINED BY DIFFERENT HOG DESCRIPTORS AND COMPETING CLASSIFIERS ON BTSC DATASET

	HOGs_nb	HOGc_nb	**HOGv_nb**	**HOGv+r**
Kernel ELM based	96.98±0.12%	98.18±0.15%	98.22±0.21%	**98.54±0.20%**
Kernel SVM based	96.82±0.18%	98.01±0.17%	98.12±0.20%	**98.49±0.19%**
ELM based	96.61±0.15%	97.80±0.16%	98.05±0.18%	**98.20±0.21%**
SVM based	96.50±0.21%	97.51±0.19%	97.67±0.16%	**98.02±0.18%**
LDA based	95.20±0.20%	96.85±0.18%	97.21±0.16%	**98.02±0.11%**

TABLE VII
EVALUATION OF DIFFERENT HOG DESCRIPTORS AND COMPETING CLASSIFIERS ON BTSC DATASET

Method	Descriptor	Maximal Recognition Rate	Average Training Time	Average Recognition Time
Kernel ELM based	HOGs_nb	97.08%	4.80s/dataset	1.00ms/frame
	HOGc_nb	98.30%	4.81s/dataset	1.00ms/frame
	HOGv_nb	98.41%	5.66s/dataset	1.29ms/frame
	HOGv+r	**98.62%**	10.3s/dataset	1.46ms/frame
Kernel SVM based	HOGs_nb	96.96%	107s/dataset	16.30ms/frame
	HOGc_nb	98.11%	107s/dataset	16.26ms/frame
	HOGv_nb	98.26%	128s/dataset	19.85ms/frame
	HOGv+r	**98.62%**	275s/dataset	28.14ms/frame
ELM based	HOGs_nb	96.72%	32.1s/dataset	1.30ms/frame
	HOGc_nb	97.95%	32.1s/dataset	1.27ms/frame
	HOGv_nb	98.18%	33.3s/dataset	1.38ms/frame
	HOGv+r	**98.38%**	50.3s/dataset	1.43ms/frame
SVM based	HOGs_nb	96.61%	68.2s/dataset	8.41ms/frame
	HOGc_nb	97.87%	68.3s/dataset	8.37ms/frame
	HOGv_nb	97.95%	85.2s/dataset	10.30ms/frame
	HOGv+r	**98.22%**	160.2s/dataset	16.30ms/frame
LDA based	HOGs_nb	95.46%	13.3s/dataset	0.55ms/frame
	HOGc_nb	97.0%	13.3s/dataset	0.55ms/frame
	HOGv_nb	97.37%	42.3s/dataset	0.59ms/frame
	HOGv+r	**98.18%**	50.2s/dataset	0.79ms/frame

method are still within the grid where their ground truth category locates. It indicates that the misclassification degree of this proposed method is lower than that of committee of CNNs at the category level.

In order to give an intuitive illustration, Fig. 8 shows all misclassified images obtained by this proposed kernel ELM-based method and committee of CNNs. It can be seen that lower resolution of some input images is the first reason causing misclassification for both methods. It accounts for more than half of misclassified images in this proposed method. Shadow and motion blurring are the second reason causing misclassification for both methods. In fact, a few images, as shown in the last line of Fig. 8(a), are so blurred that it is difficult to be recognized by humans. Another interesting point can be seen that most traffic signs with diamond shape are misclassified by the committee of CNNs while they are correctly recognized by this proposed method. It again indicates that HOGv+r descriptor used in this proposed method is powerful to represent the distinctive shapes than CNN-based methods.

TABLE VIII
OVERALL PERFORMANCE COMPARISON OF PUBLISHED METHODS ON BTSC DATASET

Method	Recognition Rate	Training Time	Recognition Time	Configuration
Kernel ELM based	**98.62%**	10.3s/dataset	1.46ms/frame	CPU: I7-4790
ELM based	98.38%	50.3s/dataset	1.42ms/frame	CPU: I7-4790
INNC+INNLP [6]	98.32%	N/A	N/A	CPU: AMD Opteron 8360SE
SRGE [5]	96.26%	N/A	N/A	CPU: I7-950

TABLE IX
AVERAGE RECOGNITION ACCURACY OBTAINED BY DIFFERENT HOG DESCRIPTORS AND COMPETING CLASSIFIERS ON REVISED MASTIF DATASET

	HOGs_nb	HOGc_nb	**HOGv_nb**	**HOGv+r**
Kernel ELM based	96.95±0.18%	97.20±0.19%	97.96±0.15%	**98.12±0.17%**
Kernel SVM based	96.72±0.17%	96.87±0.19%	97.92±0.14%	**97.98±0.15%**
ELM based	96.32±0.21%	96.98±0.15%	97.79±0.19%	**97.91±0.16%**
SVM based	96.21±0.16%	96.80±0.18%	97.70±0.16%	**97.89±0.17%**
LDA based	95.20±0.18%	96.11±0.19%	97.22±0.13%	**97.54±0.18%**

TABLE X
EVALUATION OF DIFFERENT HOG DESCRIPTORS AND COMPETING CLASSIFIERS ON REVISED MASTIF DATASET

Method	Descriptor	Maximal Recognition Rate	Average Training Time	Average Recognition Time
Kernel ELM based	HOGs_nb	97.07%	4.03s/dataset	1.02ms/frame
	HOGc_nb	97.37%	3.98s/dataset	1.04ms/frame
	HOGv_nb	98.09%	4.7s/dataset	1.23ms/frame
	HOGv+r	**98.26%**	5.4s/dataset	1.51ms/frame
Kernel SVM based	HOGs_nb	96.92%	66.2s/dataset	11.72ms/frame
	HOGc_nb	97.05%	67.1s/dataset	12.16ms/frame
	HOGv_nb	98.00%	78.3s/dataset	13.96ms/frame
	HOGv+r	**98.09%**	96.3s/dataset	15.58ms/frame
ELM based	HOGs_nb	96.52%	66.1s/dataset	3.87ms/frame
	HOGc_nb	97.20%	65.9s/dataset	3.87ms/frame
	HOGv_nb	97.90%	76.1s/dataset	5.55ms/frame
	HOGv+r	**98.0%**	103.2s/dataset	6.05ms/frame
SVM based	HOGs_nb	96.47%	42.1s/dataset	6.11ms/frame
	HOGc_nb	97.11%	41.8s/dataset	6.05ms/frame
	HOGv_nb	97.93%	54.2s/dataset	7.23ms/frame
	HOGv+r	**98.00%**	68.2s/dataset	8.91ms/frame
LDA based	HOGs_nb	95.4%	12.3s/dataset	0.53ms/frame
	HOGc_nb	96.3%	12.3s/dataset	0.53ms/frame
	HOGv_nb	97.42%	48.2s/dataset	0.55ms/frame
	HOGv+r	**97.75%**	51.2s/dataset	0.55ms/frame

D. Evaluation on BTSC and Revised MASTIF Datasets

1) Evaluation of HOG Descriptors: On BTSC and revised MASTIF datasets, background removed HOG versions, including HOGs_nb, HOGc_nb, HOGs_nb, and HOGv+r, are evaluated given each type of classifiers to show which descriptor is better. Tables VI and VII show average recognition accuracy and maximal recognition accuracy, respectively, obtained on BTSC dataset. Tables IX and X show average recognition accuracy and maximal recognition accuracy, respectively, obtained on revised MASTIF dataset.

In Tables VI and IX, each row illustrates the performance comparison among HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r given a classifier. Like GTSRB dataset, the same two facts can be found in BTSC and revised MASTIF datasets.

1) HOGv+r achieves the highest recognition accuracy for each classifier and HOGv_nb can also outperform HOGs_nb and HOGc_nb.
2) HOGv+r and HOGv_nb show the above effects not only using ELM classifier but also using other classifiers, such as SVM, kernel SVM, and LDA.

In Tables VII and X, the third column illustrates the performance comparison among HOGs_nb, HOGc_nb, HOGv_nb, and HOGv+r given each classifier. Like GTSRB dataset, the same facts can be found in BTSC and revised MASTIF datasets.

Therefore, the same conclusions can be obtained after HOG evaluation on these two datasets.

1) This proposed HOG descriptor with the combination of signed and unsigned gradients (i.e., HOGv_nb and HOGv+r) can outperform the original HOG descriptor only with signed gradients (i.e., HOGc_nb) or unsigned gradients (i.e., HOGs_nb).

Furthermore, by comparing Tables III, VII, and X, a new fact can be found that the advantage of this proposed TSR method is much more obvious in GTSRB dataset than in BTSC and revised MASTIF datasets. Based on the fact that GTSRB dataset is much larger than the other two datasets, it can be concluded that this proposed TSR method is more suitable for large data cases.

2) Evaluation of Competing Classifiers: On BTSC and revised MASTIF datasets, this proposed ELM and kernel ELM classifiers are compared against other competing classifiers (i.e., SVM, kernel SVM, and LDA) using each type of HOG descriptors including HOGs_nb, HOGc_nb, HOGv_nb, and

HOGv+r. Average and maximum of recognition accuracy, average training time, and average recognition time are used to evaluate these competing classifiers.

Tables VI and IX list the average recognition accuracy for BTSC and revised MASTIF datasets, respectively. In these two tables, each column illustrates the performance comparison among competing classifiers given an HOG descriptor. Like GTSRB dataset, the same fact can be found that kernel ELM achieves the highest recognition accuracy for each type of HOG descriptors.

Tables VII and X list the maximal recognition accuracy, average training time, and average recognition time obtained by each of competing classifiers given each type of HOG descriptors for BTSC and revised MASTIF datasets, respectively. It is important to note that the average training time and average recognition time listed in these two tables both include the time of HOG extraction. It can be seen that kernel ELM outperforms SVM, kernel SVM, and LDA for each HOG descriptor in terms of accuracy except that it has the same recognition accuracy with kernel SVM for HOGv+r.

Therefore, like GTSRB dataset, the following two facts can be found in BTSC and revised MASTIF datasets.

1) This proposed kernel ELM-based TSR method outperforms SVM, kernel SVM, and LDA-based methods in terms of recognition accuracy and training time.
2) Although kernel ELM-based TSR method takes a little more recognition time than LDA-based method, its recognition accuracy is higher than LDA.

3) Overall Performance Comparison: For BTSC dataset, there are only two published methods found, denoted as INNC+INNLP [6] and SRGE [5]. Information of these published methods can be seen in Section VI-C4. Results of these methods are obtained from their published papers.

Table VIII lists the recognition accuracy of this proposed method and others. It can be seen that this proposed kernel ELM-based TSR method outperforms these two published methods in terms of accuracy.

VII. Discussion

Above experimental results have shown that this proposed combination of hand-designed HOGv descriptor and ELM-based classifier has the comparable recognition performance with DNN-based methods which automatically learn features. However, this proposed method can outperform DNN-based methods in terms of computational efficiency. This fact can be further explained by computational complexity analysis for recognition process.

The computational complexity of this proposed method can be approximately represented as $\mathcal{O}(nZ) + \mathcal{O}(L)$, where Z denotes the number of image pixels, n denotes the steps required for HOGv extraction, and L denotes the number of hidden nodes in ELM. In this proposed method, $n = 5$ and $L \ll Z$. The computational complexity of CNN-based methods can be approximately represented as $\sum_l \mathcal{O}(n_l Z_l)$, where l denotes the index of each network level, n_l denotes the number of filters at network level l, and Z_l denotes the number of image pixels at level l. It can be seen that the complexity of this proposed method only includes two parts while that of CNN-based methods includes multiple parts due to their multilayer structure. Considering the first part of two types of methods, $n \ll n_1$ and $Z = Z_1$. Considering the second part, $L \ll n_2 Z_2$. It can be seen that the computational complexity of this proposed method is much lower than that of CNN-based methods. It can be concluded that this proposed method can obtain better balance between recognition accuracy and computational efficiency compared against other methods, e.g., CNN-based methods.

VIII. Conclusion

This paper proposes a computationally efficient method for TSR. This method presents a variant HOG descriptor with signed and unsigned gradients for building feature representation and develops a single-hidden-layer neural network that is trained using ELM algorithm for classification. Three benchmark datasets (i.e., GTSRB, BTSC, and revised MASTIF) have been used to evaluate this proposed method. Experimental results in three datasets have shown that this proposed method outperforms most of state-of-the-art methods in terms of recognition accuracy and computational efficiency. Compared with DNN-based methods which run with high-performance GPUs, this proposed method costs much less during training and recognition under a standard PC configuration. So high recognition efficiency makes it very promising for real-time applications. Meanwhile, the training efficiency makes it easy to train large volume of data and it also allows online real-time update. Furthermore, experimental results have shown that this proposed method has little dependence on parameter tuning.

Future work includes the learning of number of hidden nodes and extending this ELM-based classifier for traffic sign detection.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, 2005, pp. 886–893.

[2] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A hierarchical method for traffic sign classification with support vector machines," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Dallas, TX, USA, 2013, pp. 1–6.

[3] F. Zaklouta, B. Stanciulescu, and O. Hamdoun, "Traffic sign classification using K-d trees and random forests," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, San Diego, CA, USA, 2011, pp. 2151–2155.

[4] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.

[5] K. Lu, Z. Ding, and S. Ge, "Sparse-representation-based graph embedding for traffic sign recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1515–1524, Dec. 2012.

[6] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool, "Traffic sign recognition—How far are we from the solution?" in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Dallas, TX, USA, 2013, pp. 1–8.

[7] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs computer: Benchmarking machine learning algorithms for traffic sign recognition," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, San Jose, CA, USA, 2011, pp. 323–332.

[8] 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 Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.

[9] S. Maldonado-Bascón, J. Acevedo-Rodríguez, S. Lafuente-Arroyo, A. Fernndez-Caballero, and F. López-Ferreras, "An optimization on pictogram identification for the road-sign recognition task using SVMs," *Comput. Vis. Image Understand.*, vol. 114, no. 3, pp. 373–383, 2010.

[10] D. C. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 3642–3649.

[11] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, San Jose, CA, USA, 2011, pp. 2809–2813.

[12] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine theory and application," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramana, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[15] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[16] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. C. Soh, "Stacked extreme learning machines," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2013–2025, Sep. 2015.

[17] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Comput. Intell. Mag.*, vol. 10, no. 2, pp. 18–29, May 2015.

[18] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Dec. 2013.

[19] R. Janssen, W. Ritter, F. Stein, and S. Ott, "Hybrid approach for traffic sign recognition," in *Proc. Intell. Vehicles Symp.*, Tokyo, Japan, 1993, pp. 390–395.

[20] S. Tang and L.-L. Huang, "Traffic sign recognition using complementary features," in *Proc. Asian Conf. Pattern Recognit.*, Naha, Japan, 2013, pp. 210–214.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[22] M. Takaki and H. Fujiyoshi, "Traffic sign recognition using SIFT features," *IEEJ Trans. Electron. Inf. Syst.*, vol. 129, no. 5, pp. 824–831, 2009.

[23] A. Ihara, H. Fujiyoshi, M. Takaki, H. Kumon, and Y. Tamatsu, "Improvement in the accuracy of matching by different feature subspaces in traffic sign recognition," *IEEJ Trans. Electron. Inf. Syst.*, vol. 129, no. 5, pp. 893–900, 2009.

[24] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. America B Opt. Phys.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[25] F. Măriuţ, C. Foşalău, M. Avila, and D. Petrişor, "Detection and recognition of traffic signs using Gabor filters," in *Proc. Int. Conf. Telecommun. Signal Process.*, Budapest, Hungary, 2011, pp. 554–558.

[26] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[27] X. Yuan, X. Hao, H. Chen, and X. Wei, "Robust traffic sign recognition based on color global and local oriented edge magnitude patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1466–1477, Aug. 2014.

[28] Y. Zhu, X. Wang, C. Yao, and X. Bai, "Traffic sign classification using two-layer image representation," in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, VIC, Australia, 2013, pp. 3755–3759.

[29] J. Wang *et al.*, "Locality-constrained linear coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3360–3367.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, 2006, pp. 2169–2178.

[31] M. S. Prieto and A. R. Allen, "Using self-organising maps in the detection and recognition of road signs," *Image Vis. Comput.*, vol. 27, no. 6, pp. 673–683, 2009.

[32] Z. Huang, Y. Yu, and J. Gu, "A novel method for traffic sign recognition based on extreme learning machine," in *Proc. World Congr. Intell. Control Autom.*, Shenyang, China, 2014, pp. 1451–1456.

[33] Y. Zeng, X. Xu, Y. Fang, and K. Zhao, "Traffic sign recognition using deep convolutional networks and extreme learning machine," in *Intelligence Science and Big Data Engineering. Image and Video Data Engineering* (LNCS 9242). Cham, Switzerland: Springer, 2015, pp. 272–280.

[34] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3485–3492.

[35] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[36] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3056–3062, 2007.

[37] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, 2007.

[38] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[39] R. Timofte and L. V. Gool, "Sparse representation based projections," in *Proc. Brit. Conf. Mach. Vis.*, Dundee, U.K., 2011, pp. 61–72.

[40] I. Filković. *Traffic Sign Localization and Classification Methods: An Overview*. [Online]. Available: http://www.fer.unizg.hr/_download/repository/KDI,_Ivan_Filkovic.pdf.

[41] S. Šegvic *et al.*, "A computer vision assisted geoinformation inventory for traffic infrastructure," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Funchal, Portugal, 2010, pp. 66–73.

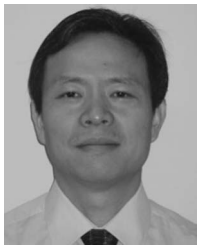
[42] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

Zhiyong Huang received the bachelor's degree in computer science and technology from Fuzhou University, Fuzhou, China, in 2014, where he is currently pursuing the master's degree.

His current research interests include computer vision and machine learning.

Yuanlong Yu received the Ph.D. degree in electrical engineering from the Memorial University of Newfoundland, St. John's, NL, Canada, in 2010.

Since 2011, he has been a Post-Doctoral Fellow with the Memorial University of Newfoundland and Dalhousie University, Halifax, NS, Canada. Since 2013, he has been a Professor with Fuzhou University, Fuzhou, China. His current research interests include computer vision, machine learning, visual attention, autonomous mental development, and cognitive robotics.



Jason Gu received the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2000.

He is a Professor with Dalhousie University, Halifax, NS, Canada. His current research interests include robotics, biomedical engineering, and control.

Huaping Liu received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2004.

He is an Associate Professor with Tsinghua University. His current research interests include intelligent control and robotics.