



Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science &amp; Engineering and Electrical Engineering

www.scientiairanica.com



# Recognizing traffic signs using flexible Discrete Cosine Transform (DCT) grid

S.K. Noon<sup>a</sup>, K. Javed<sup>b</sup>, A. Mannan<sup>a,\*</sup> and H.A. Babri<sup>b</sup>

a. Department of Electrical Engineering, NFC Institute of Engineering and Technology, Multan, Pakistan.

b. Department of Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan.

Received 30 June 2015; received in revised form 29 November 2015; accepted 20 February 2016

## KEYWORDS

Feature selection;  
Traffic sign  
recognition;  
Feature extraction;  
Discrete cosine  
transform;  
Support vector  
machine.

**Abstract.** Traffic sign recognition can be performed in two phases of detection and recognition; detection deals with sensing a traffic sign in real-world image or video frame while recognition is about reading its contents. A traffic signs database may contain samples with varying font sizes and styles used for printing the interior of a traffic sign and the contents may also be shifted away from the center of gravity. In this paper, we utilize the energy compaction property of Discrete Cosine Transform (DCT) to propose a Traffic Sign Recognition (TSR) system, which can generate invariant features for varying font styles and scaled up, scaled down, and translated contents of a sign. Experiments on synthetic and real-world images datasets show that the features generated by our proposed method have great intra-class similarity and inter-class variation. We have also shown that our proposed method outperforms Eigen based recognition method [1] and is comparable with the Histogram of Oriented Gradient (HOG) approach [2] using Support Vector Machine (SVM) classifier.

© 2017 Sharif University of Technology. All rights reserved.

## 1. Introduction

Taking care of speed traffic signs ensures a smooth and risk-free drive. Today, with top quality highways and speedy vehicles, there is a more stringent need for abiding by traffic rules than ever. Therefore, vehicles being able to detect a traffic board and inform the driver can be very helpful in reducing road accidents [3,4].

As shown in Figure 1, there are two broad categories of traffic signs in the world; United States (US) based (Figure 1(a)) and United Kingdom (UK) based (Figure 1(b)) [5]. US based traffic signs are rectangular with white background and black foreground. UK inspired traffic signs, prevalent in most countries across

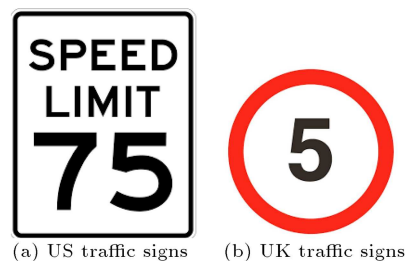


Figure 1. Samples of US and UK traffic signs.

the world including Pakistan, are generally red bordered and are either mandatory (circular) or warning (triangular) signs [6]. On violation of mandatory signs, e.g. speed limit sign, a fine is imposed, whereas warning signs are to inform the driver for example of a U-turn ahead.

Recognizing a speed sign is challenging because factors such as weather [1,7], varying illumination [8,9], signs being deteriorated due to poor maintenance (at some places) and occlusion [10,11], etc. contribute

\*. Corresponding author. Tel.: 0092 619220012;

Fax: 0092 619220018

E-mail addresses: seroshkarim@nfciet.edu.pk (S.K. Noon);

kashif.javed@uet.edu.pk (K. Javed); mannan@nfciet.edu.pk

(A. Mannan); babri@uet.edu.pk (H.A. Babri)

to making the task difficult. Most of the literature on traffic sign recognition uses Hue Saturation Value (HSV) color space [1,9,12] with some range of Hue and Saturation to detect red color while some use predefined ranges in RGB color space to segment red objects [13,14]. Some authors have attempted to take on the problem by detecting shape of the sign first [9]. A number of techniques like ANN [5], template matching [15], and principal component analysis [1] have been used to recognize type of sign.

In real-world situations, size and type of font and width of line or text used to draw interior of a sign may vary from one sign to another. Moreover, the interior stuff may be translated up, down, backward, or forward. These issues can easily affect the performance of a traffic sign recognition system. We, in this paper, present a simple yet effective technique to correctly read the contents of signs affected by one or more such issues. We use energy compaction property of Discrete Cosine Transform (DCT) [16] to generate invariant feature for traffic signs. Since signs captured closely appear greater in size than snapshots taken from a distance, we, irrespective of the size of the sign in a scene, use a flexible grid of size  $8 \times 8$  and apply DCT to each block separately. Since energy in the transformed image blocks is concentrated in the top left corner [16], we pick the top three components (DC and the first two ACs) in each block and concatenate them to form a feature vector of fixed dimension. For a variety of synthetic and real-world images, we see an amazing intra-class similarity in the feature vectors generated by different images containing the same sign. Moreover, the feature vectors generated from images containing different signs are considerably different from each other, hence, having desirable inter-class variation. These two characteristics of the feature vector make our proposed algorithm suitable to identify contents of a traffic sign in diverse situations. In order to judge the performance of our proposed algorithm, we used Support Vector Machine (SVM) classifier [17] in the experiments for training and testing purposes.

The remainder of the this paper is organized as follows: Section 2 describes the work related to reading the interior/content of a traffic sign; Section 3 introduces our proposed flexible DCT grid based traffic sign recognition system; Section 4 mentions the data and classifiers used to evaluate different traffic sign recognition algorithms; Section 5 shows results; and, finally, we draw conclusion in Section 6.

## 2. Related work

Recognizing the contents of a traffic sign is an important phase of a traffic sign recognition system. The content can be a number of two or three digits (for mandatory speed limit signs) or various shapes (like

arrows for left or right turn signs). Researchers have been experimenting with various techniques to correctly and reliably read the interior of a sign detected in a scene [18,19].

Fleyeh and Davami [1] segment input image by using Hue to detect red color in conjunction with saturation to avoid achromatic area. The sub-image corresponding to the location of traffic sign is processed using PCA to extract the invariant feature corresponding to each sign. Their proposed system can effectively identify signs in poor weather conditions but is not good when it comes to reading signs with translated contents. Barnas and Zelinsky [11] propose a shape detection algorithm based on radial symmetry algorithm based on circular Hough transforms by arguing that color may vary with changing light conditions but shape remains stable. They use cross correlation [20] to recognize the contents of a sign and provide promising results on images collected around their campus. However, their system performs poorly for occluded signs. Bascon et al. [9] use RGB color thresholding for detection and SVM for recognition of traffic signs on Spanish roads. They provide good results in the bright sunshine of the day but RGB color thresholding is not good for poor lighting conditions. Hoferlin and Zimmermann [21] propose a Scale Invariant Feature Transform (SIFT) based approach to recognize circular traffic signs as per their contents. Hoferlin's work is limited to recognizing only circular signs, which in most parts of the world are used to enforce speed limits most of the time. Varan et al. [22] present a template matching based automated traffic sign recognition system generating invariant features for translation, scale, rotation, weather conditions, and partial occlusion. They provide promising results, but the template matching procedure is computationally expensive.

Greenhalgh and Mirmehdi [2] use Histogram of Oriented Gradient (HOG) technique to detect the contents of self-collected traffic signs in United Kingdom with reasonably good accuracy. HOG was originally proposed by Dalal and Triggs [23] for human detection in real-world images. Ghica et al. [24] use multi-layer perception Artificial Neural Network (ANN) [25] to recognize the interior of a traffic sign. Mirmehdi's work is state of the art in the field of image processing based traffic sign recognition, but is slightly computational resource hungry.

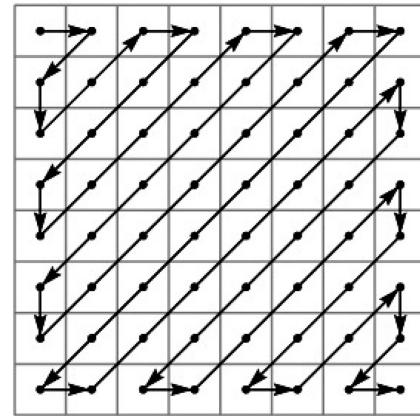
## 3. Our proposed algorithm: Flexible DCT Grid based feature extraction (FDG)

There are a number of techniques used for detection of traffic signs in real-world images [19,26] using RGB and HSI color spaces [16] to segment red bordered circular and triangular signs. Since Hue component in HSI color

space carries the information of color in the image, we threshold this plane to extract closed red bordered shapes. This is followed by a morphological hole filling operation [16] to fill the interior of the detected shape. The filled image is subtracted from the thresholded image and the difference is the interior of the sign as a blob. This small portion is taken out of the scene to apply our proposed flexible DCT grid based recognition technique. The process of sign segmentation is shown in Figure 2.

Here, our focus is on generating invariant features for the traffic sign extracted in Figure 2(e). We use energy compaction property of Discrete Cosine Transform (DCT) to generate features for the detected traffic signs having high inter-class variation and intra-class similarity. DCT has been previously used for recognition purposes in Content Based Image Retrieval (CBIR) [27] and binary object recognition [28] under elastic deformations and noise. DCT is a variant of Fourier Transform [29] meant for digital signals/images. It is famous for its property of pulling the energy of a two-dimensional signal/image in its upper left corner. It is evident from Eq. (1) that the top left DCT coefficient is the sum of the values of all pixels divided by the square root of the size of the image and is called the DC component. All other pixels in the transformed image are AC components of increasing frequency. DCT is widely used in JPEG image compression [30] where the pixels in the transformed image are picked in a zigzag form shown in Figure 3.

For the task of generating invariant features for

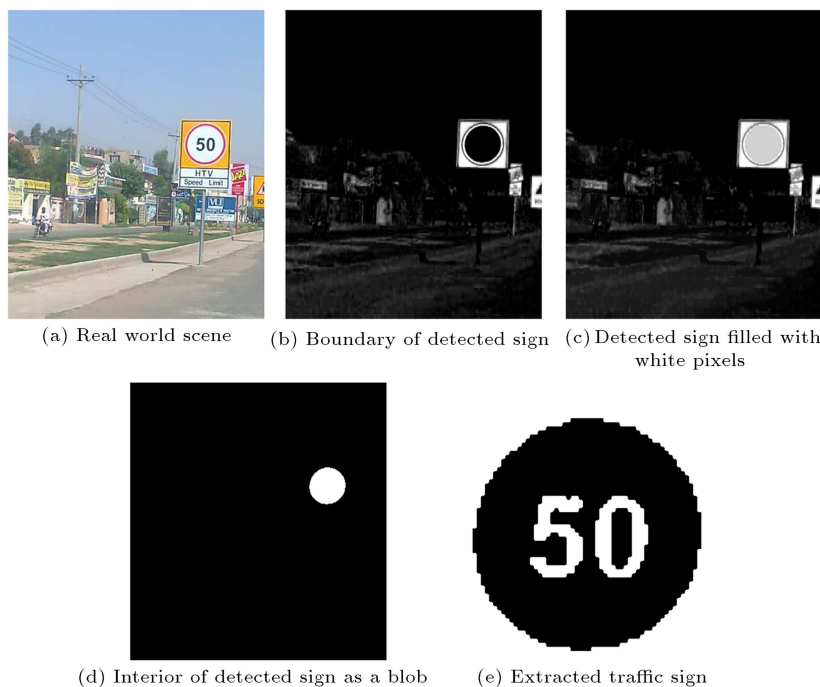


**Figure 3.** Selecting DCT components in zigzag form by JPEG.

traffic signs, we use the first three components of the DCT transformed image, i.e. DC, the first AC, and the second AC [28]. The general expression to calculate DCT of a two-dimensional matrix,  $f(i, j)$ , of size  $M \times N$  is given in Eq. (1) and the three components are calculated mathematically as shown in Eq. (2) through Eq. (4):

$$F(u, v) = \alpha_u \alpha_v \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) \cos \frac{\pi(2i+1)u}{2M} \cos \frac{\pi(2j+1)v}{2N}, \quad (1)$$

while  $0 \leq u \leq M-1$ ,  $0 \leq v \leq N-1$  and:



**Figure 2.** Detection of a red bordered speed limit sign in a real-world scene.

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{M}}, & \text{if } u = 0 \\ \sqrt{\frac{2}{M}}, & \text{otherwise} \end{cases}$$

$$\alpha_v = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } v = 0 \\ \sqrt{\frac{2}{N}}, & \text{otherwise} \end{cases}$$

$$F(0,0) = \alpha_0 \alpha_0 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cos(0) \cos(0), \quad (2)$$

$$F(0,1) = \alpha_0 \alpha_1 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cos(0) \cos\left[\frac{\pi(2j+1)}{2N}\right], \quad (3)$$

$$F(1,0) = \alpha_1 \alpha_0 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cos\left[\frac{\pi(2i+1)}{2M}\right] \cos(0). \quad (4)$$

The first and second AC components (Eqs. (3) and (4)) correspond to slowest variations in vertical and horizontal directions, respectively, in the input matrix. We concatenate these three components of the transform in each block to form the feature vector. Since the grid is flexible and is always  $8 \times 8$  irrespective of the size of the image, the size of feature vector remains 192 for all cases. In order to cater for scale invariance (i.e., traffic signs captured closely appear larger in size than the signs captured from a distance), we normalize the feature vectors by dividing each value by the maximum value in the feature vector. This keeps all values of the 192-dimensional feature vector between zero and one.

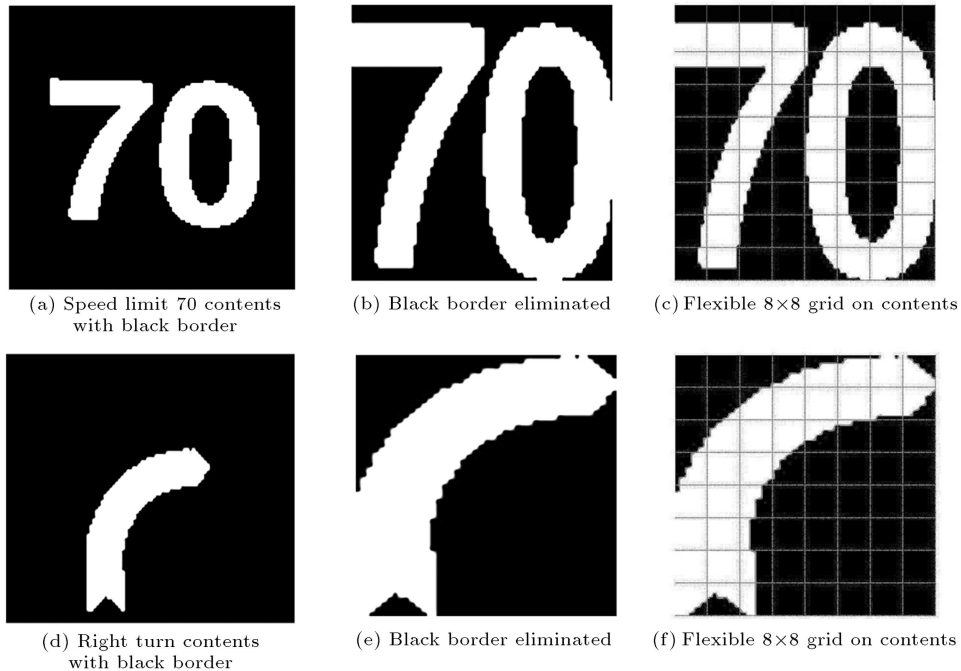
### 3.1. Preprocessing

As shown in Figure 2(a), the extracted sign has a black border around, which should be removed before generating its feature vector. Moreover, the sign may be translated into any of the four possible directions; black border removal also caters for this issue after it is segmented from its background. Figure 4 shows how translated contents are dealt with by removing useless black borders around the contents and then dividing the rest of the images in  $8 \times 8$  grid so that DCT can be applied to each of the 64 blocks.

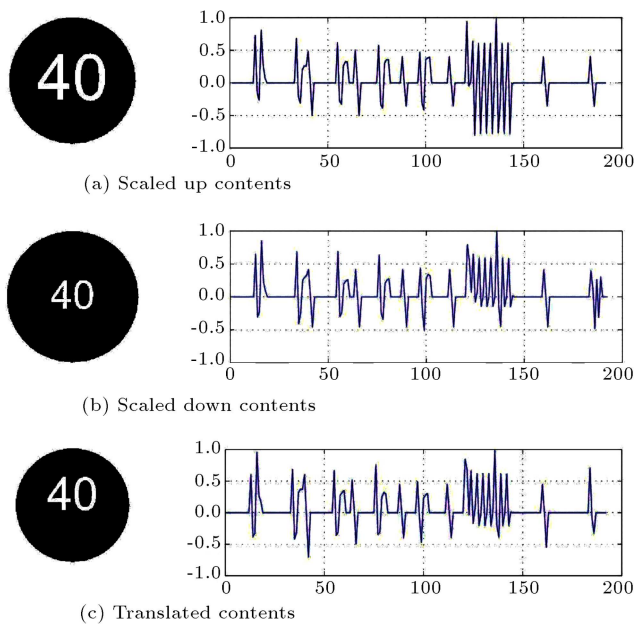
### 3.2. Generating feature vectors

After removing the black border and dividing the image into 64 boxes of equal area, DCT is applied to each block and since the energy is concentrated in the top left corner, the top three components are taken, i.e. DC, the first AC, and the second AC, from each block. This gives a 192-dimensional feature vector for the traffic sign.

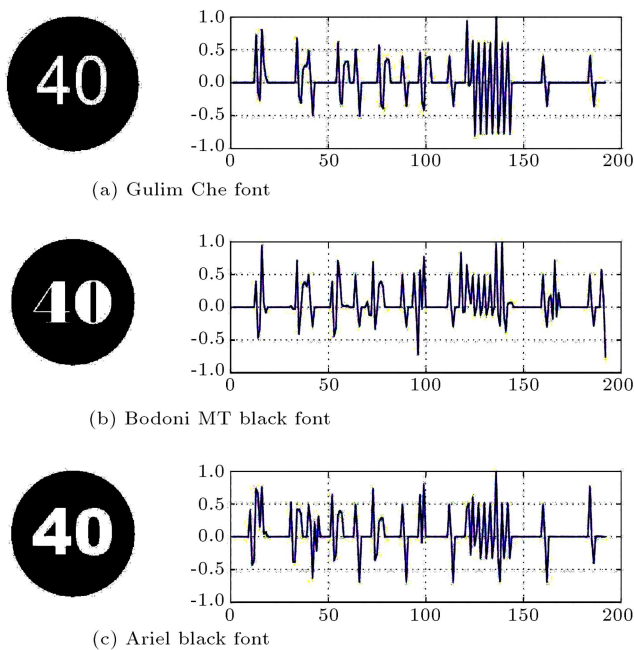
Various versions of traffic signs with scaled and translated contents from synthetic dataset are shown in Figure 5 on the left and their corresponding feature vectors are shown on the right. Figure 5(a) shows a speed limit 40 traffic sign with contents in a fairly large font size and its feature vector; Figure 5(b) shows the same sign with contents printed in the same font style but with a reduced font size; the generated feature vector is very close to as shown in Figure 5(a). The feature vector remains invariant even if the contents are translated upward (moving the content up within the rim) as shown in Figure 5(c). This shows that



**Figure 4.** Black border removal and applying of flexible grid on the contents of a sample traffic sign.



**Figure 5.** Generating invariant features from a traffic sign with contents scaled up, scaled down, and translated.



**Figure 6.** Generating invariant features from a traffic sign with contents written in different fonts.

the proposed algorithm is invariant to scaling and translation of the contents of a traffic sign.

Another interesting characteristic of the proposed method is its invariance to various known font styles. In the synthetic dataset, we printed the contents of various traffic signs in different fonts and found the feature vectors invariant for ten different writing styles. Figure 6 shows the invariance of feature vector for three well known font styles. It is clear from Figure 6(a),

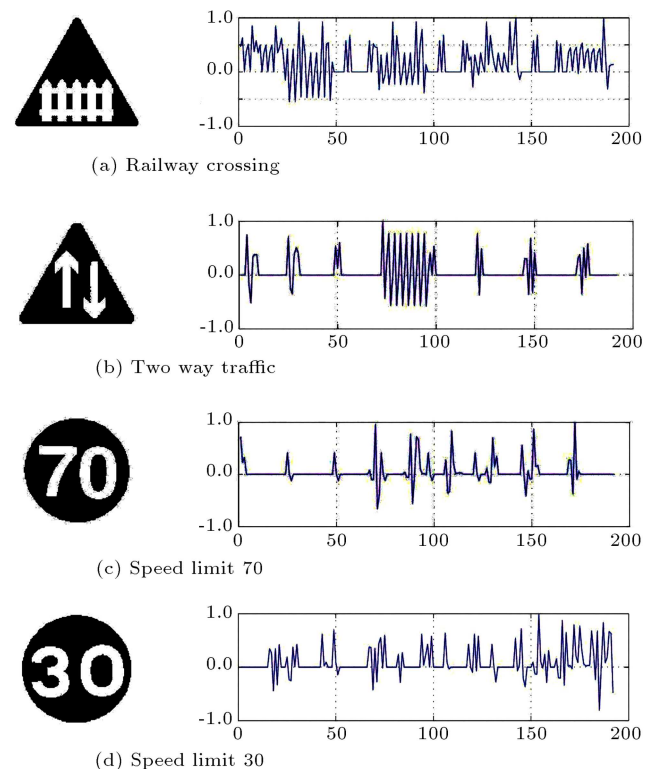
(b), and (c) that the 192-dimensional feature vectors generated through our proposed algorithm are very close for Gulim Che, Bodoni MT Black, and Ariel Black fonts.

### 3.3. Inter-class variation

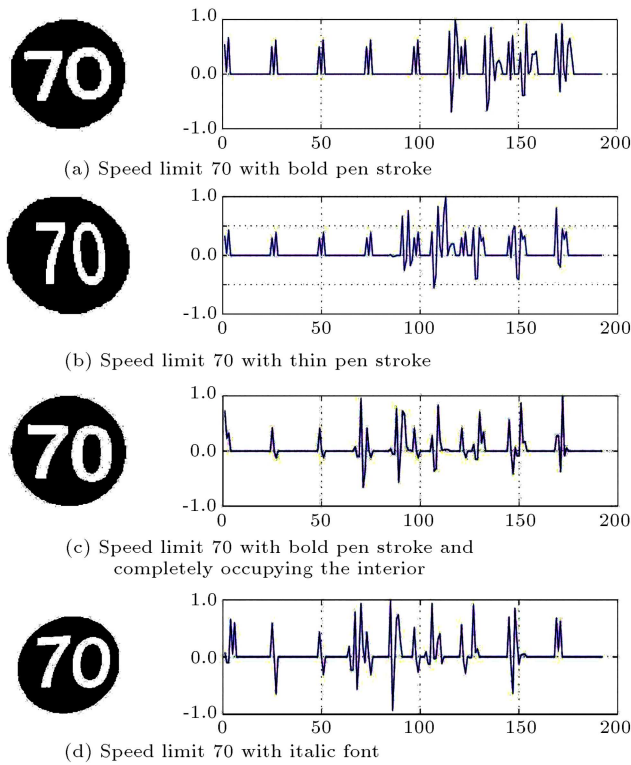
A desirable characteristic of a feature extraction technique is that the vectors generated for different classes, i.e. traffic signs, must be significantly different from each other. This helps in distinguishing a traffic sign in the presence of instances from various other classes. Figure 7 shows instances from circular and triangular traffic signs taken from real-world dataset. We can see that the feature vectors generated for railway crossing (Figure 7(a)) and two-way traffic (Figure 7(b)) show very different patterns. Similarly, Figure 7(c) and (d) show very weak correlation between the generated features. Overall analysis of all the four instances from different classes in Figure 7 reveals that our proposed algorithm produces significantly different features/signatures for different types/classes of traffic signs.

### 3.4. Intra-class similarity

Feature extraction using our proposed Flexible DCT Grid (FDG) based technique shows that various instances of the same class generate very similar features. Figure 8 shows the feature vectors generated for four instances of speed limit 70 sign taken from real-world dataset. Figure 8(a) shows the sign with properly filled



**Figure 7.** Inter-class variation across various traffic signs.



**Figure 8.** Intra-class similarity across various samples of a traffic sign.

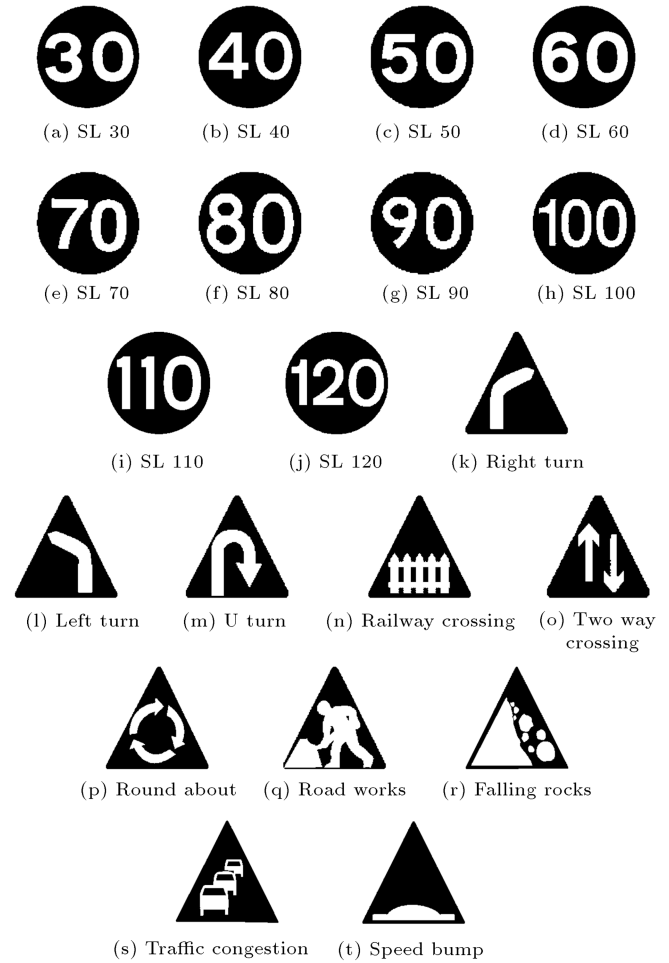
contents and its feature vector whereas Figure 8(b) shows another instance of the same class, but with different size and type of font (length is more than the width). The traffic sign shown in Figure 8(c) also belongs to the same class, but is slightly different in appearance and, finally, Figure 8(d) shows a sample of the same speed limit sign with board a bit tilted. All the four feature vectors shown in Figure 8 show that for apparently different looking instances of the same class, the feature vectors are highly correlated.

This shows that the proposed feature selection algorithm produces features with great intra-class similarity and inter-class variation.

## 4. Experiments

### 4.1. Datasets

We used two datasets for experiments, namely, synthetic and real-world datasets, both having twenty classes. Synthetic dataset contains 1600 instances with 80 samples per class. Images in each class contain scaled up, scaled down, and translated contents printed with 10 different fonts. Real-world dataset contains 1252 images with almost 63 instances per class. The scenes are captured by the authors from various Pakistani roads. The data is a random mixture of scaled, skewed, and variable-rim-width contents printed with both standard and non-standard fonts. Representative images of each class are shown in Figure 9.



**Figure 9.** Samples of traffic signs used for experiments.

In order to investigate error rate, precision, and recall, both datasets are divided as follows: randomly chosen 70% of the images are used for training and the remaining 30% are used for testing using Support Vector Machine (SVM) [31] classifier. SVM is a widely used classifier famous for its good performance and simplicity of implementation. Originally, SVM was intended to distinguish between only two classes, but it can be extended to handle more than two classes through one-versus-all technique, i.e. at a time, one class is considered as positive and the rest of the classes are considered as negative. The process is repeated equal to the number of classes in the data. We use the SVM implementation given in MATLAB [32].

Performance analysis of the synthetic dataset, real-world dataset, and their combination is performed through three parameters, namely, error rate, precision, and recall. These are defined in Eqs. (5) to (7):

$$\text{Error\_rate} = \frac{\text{false\_positives}}{\text{false\_positives} + \text{true\_negatives}}, \quad (5)$$

$$\text{Precision} = \frac{\text{true\_positives}}{\text{true\_positives} + \text{false\_positives}}, \quad (6)$$



$$\text{Recall} = \frac{\text{true\_positives}}{\text{true\_positives} + \text{false\_negatives}}. \quad (7)$$

## 5. Results

Synthetic and real-world datasets are divided into training and test subsets as mentioned in Section 4. Initially, error rate using SVM classifier is calculated for synthetic and real-world datasets. Then, both datasets are combined and the experiment is repeated. In order to analyze the utility of our proposed approach, we compare it with two other techniques, i.e. Eigen based [1] and HOG based [2] feature extraction. For comparisons with Fleyeh's work [1], which applies PCA to extracted images and uses the top twenty Eigenvec-tors as features.

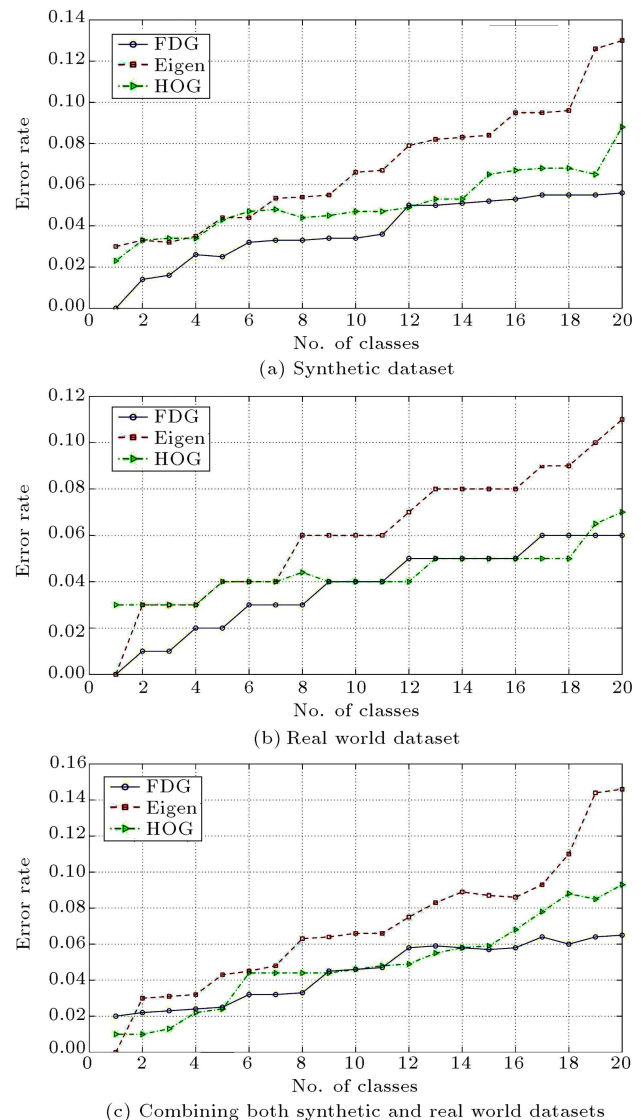
Figure 10 shows error rates of synthetic, real-world, and their combination versus increasing number of classes for all the three methods of feature extraction. A common trend of increasing error rate versus increasing number of classes can be clearly observed.

As shown in Figure 10(a), the error rate of our proposed method (FDG) is the least throughout. Histogram of Oriented Gradients (HOG) based method is comparable in the middle, but the supremacy of our proposed method becomes more evident as the number of classes reach up to 20. The third method, i.e. Eigen based feature extraction, is the poorest. A disadvantage of Eigen based method is that we have to empirically select the number of top Eigenvec-tors that give the best accuracy under given circum-stances [33].

The pattern also repeats itself for real-world dataset. However, as shown in Figure 10(b) for 12, 17, and 18 classes, HOG based feature extraction approach is the best of all. But, for all other classes, our proposed approach outperforms the other two. Once again, the Eigen based method exhibits the highest error rate throughout the experiment.

Figure 10(c) shows that for the first three classes, HOG based approach is the most accurate for the combination of synthetic and real-world datasets. Our proposed approach is the best almost throughout the experiment and Eigen based method is also the worst even for this combined dataset.

Table 1 shows the individual precision and recall of the three methods for various traffic signs from synthetic dataset. The table also shows the traffic signs most commonly misjudged by each method in each class. For road works, falling rocks, and traffic congestion signs, precision of our proposed FDG algo-rithm is very high; in addition, for falling rocks, our method recalls all relevant signs quite satisfactorily. In the last row, we average both precision and recall results achieved by each method. Table 1 shows that



**Figure 10.** Error rate of synthetic and real-world datasets versus increasing number of classes.

our proposed method achieves the highest precision and recall for synthetic dataset.

For real-world dataset (as shown in Table 2), our proposed method is the best in terms of both precision and recall. HOG based method has a reasonable overall precision and for certain individual signs, e.g. U-turn, it has quite high precision, i.e. 0.968. Eigen based method comes the last in terms of both precision and recall.

In order to further investigate the performance of the three competing methods, we joined the two datasets and determined the precision and recall of individual signs in the combined set. As shown in Table 3, our proposed method achieved overall precision and recall greater than what were achieved in cases of synthetic and real-world datasets. However, recall was a bit smaller than what we achieved in case of synthetic dataset and comparable to the recall achieved in case of real-world dataset.

**Table 1.** Precision and recall of individual traffic signs in synthetic dataset.

Traffic sign	FDG			Eigen [1]			HOG [2]		
	Pre.*	Recall	Commonly misclassified as	Pre.*	Recall	Commonly misclassified as	Pre.*	Recall	Commonly misclassified as
Speed limit 30	.920	.932	Speed limit 40	.905	.915	Speed limit 50	.913	.913	Speed limit 40
Speed limit 40	.942	.924	Speed limit 110	.895	.868	Speed limit 110	.932	.994	Speed limit 60
Speed limit 50	.962	.974	Speed limit 30	.885	.976	Speed limit 30	.944	.913	Speed limit 100
Speed limit 60	.944	.988	Speed limit 40	.845	.982	Speed limit 80	.935	.901	Speed limit 30
Speed limit 70	.920	.923	Speed limit 90	.905	.795	Speed limit 90	.943	.991	Speed limit 90
Speed limit 80	.972	.919	Speed limit 60	.935	.815	Speed limit 60	.953	.803	Speed limit 60
Speed limit 90	.982	.949	Speed limit 70	.905	.968	Speed limit 110	.953	.862	Speed limit 100
Speed limit 100	.955	.904	Speed limit 40	.889	.885	Speed limit 110	.943	.976	Speed limit 110
Speed limit 110	.990	.906	Speed limit 100	.915	.799	Speed limit 40	.894	.981	Speed limit 40
Speed limit 120	.977	.953	Speed limit 110	.944	.948	Speed limit 40	.888	.813	Speed limit 100
Right turn	.924	.943	U turn	.934	.943	U turn	.900	.947	U turn
Left turn	.992	.993	Left turn	.905	.945	Round about	.955	.844	Round about
U turn	.962	.945	Right turn	.850	.987	Right turn	.950	.804	Right turn
Railway crossing	.942	.994	Round about	.882	.989	U turn	.934	.984	Right turn
Two way traffic	.974	.895	U turn	.955	.753	U turn	.953	.954	U turn
Round about	.992	.999	Traffic congestion	.897	.899	Falling rocks	.900	.933	Right turn
Road works	.993	.989	Falling rocks	.889	.789	Railway crossing	.888	.992	Speed bump
Falling rocks	.982	.980	Speed bump	.900	.773	Road works	.876	.893	Road works
Traffic congestion	.972	.949	Speed bump	.804	.714	Speed bump	.888	.879	Falling rocks
Speed bump	.925	.948	Falling rocks	.844	.774	Falling rocks	.903	.808	Falling rocks
<b>Average</b>	.961	.950		.894	.875		.922	.909	

\*Pre.: Precision.

**Table 2.** Precision and recall of individual traffic signs in real-world dataset.

Traffic sign	FDG			Eigen [1]			HOG [2]		
	Pre.*	Recall	Commonly misclassified as	Pre.*	Recall	Commonly misclassified as	Pre.*	Recall	Commonly misclassified as
Speed limit 30	.985	.929	Speed limit 40	.923	.919	Speed limit 50	.904	.913	Speed limit 40
Speed limit 40	.985	.962	Speed limit 110	.910	.845	Speed limit 110	.946	.993	Speed limit 100
Speed limit 50	.965	.893	Speed limit 30	.839	.983	Speed limit 60	.955	.915	Speed limit 100
Speed limit 60	.970	.904	Speed limit 80	.909	.988	Speed limit 80	.957	.906	Speed limit 30
Speed limit 70	.989	.988	Speed limit 90	.822	.798	Speed limit 90	.943	.992	Speed limit 90
Speed limit 80	.929	.980	Speed limit 60	.891	.819	Speed limit 60	.933	.804	Speed limit 60
Speed limit 90	.947	.902	Speed limit 70	.889	.968	Speed limit 110	.956	.862	Speed limit 100
Speed limit 100	.949	.895	Speed limit 40	.918	.885	Speed limit 110	.937	.988	Speed limit 110
Speed limit 110	.966	.898	Speed limit 100	.834	.799	Speed limit 40	.943	.992	Speed limit 40
Speed limit 120	.970	.969	Speed limit 110	.891	.944	Speed limit 40	.958	.881	Speed limit 100
Right turn	.949	.974	U turn	.900	.943	U turn	.969	.948	U turn
Left turn	.960	.896	Round about	.909	.949	Round about	.930	.842	Round about
U turn	.989	.944	Right turn	.899	.988	Right turn	.968	.809	Right turn
Railway crossing	.980	.909	Falling rocks	.891	.985	U turn	.943	.933	Speed bump
Two way traffic	.989	.885	U turn	.899	.756	U turn	.964	.958	U turn
Round about	.956	.946	Traffic congestion	.900	.898	Falling rocks	.943	.958	Right turn
Road works	.959	.979	Speed bumps	.803	.784	Railway crossing	.956	.933	Falling Rocks
Falling rocks	.949	.965	Road works	.806	.778	Road works	.948	.893	Speed bumps
Traffic congestion	.989	.979	Speed bump	.816	.714	Speed bump	.936	.887	Falling rocks
Speed bump	.929	.937	Falling rocks	.825	.789	Falling rocks	.940	.802	Falling rocks
<b>Average</b>	.964	.936		.873	.876		.946	.910	

\*Pre.: Precision.



**Table 3.** Precision and recall of individual traffic signs in synthetic and real-world datasets.

Traffic sign	FDG			Eigen [1]			HOG [2]		
	Pre.*	Recall	Commonly misclassified as	Pre.*	Recall	Commonly misclassified as	Pre.*	Recall	Commonly misclassified as
Speed limit 30	.970	.947	Speed limit 40	.954	.934	Speed limit 50	.945	.974	Speed limit 40
Speed limit 40	.987	.945	Speed limit 110	.914	.865	Speed limit 110	.944	.905	Speed limit 100
Speed limit 50	.966	.929	Speed limit 30	.923	.823	Speed limit 100	.976	.907	Speed limit 100
Speed limit 60	.987	.907	Speed limit 80	.934	.943	Speed limit 80	.965	.916	Speed limit 30
Speed limit 70	.985	.986	Speed limit 90	.939	.794	Speed limit 90	.895	.985	Speed limit 90
Speed limit 80	.966	.894	Speed limit 60	.948	.835	Speed limit 60	.893	.844	Speed limit 60
Speed limit 90	.987	.904	Speed limit 70	.956	.933	Speed limit 110	.943	.853	Speed limit 100
Speed limit 100	.978	.916	Speed limit 40	.939	.837	Speed limit 110	.967	.985	Speed limit 110
Speed limit 110	.969	.989	Speed limit 100	.927	.756	Speed limit 40	.956	.947	Speed limit 40
Speed limit 120	.997	.963	Speed limit 110	.948	.975	Speed limit 40	.934	.803	Speed limit 100
Right turn	.970	.892	U turn	.935	.924	U turn	.971	.988	U turn
Left turn	.999	.897	Round about	.963	.927	Round about	.964	.812	Round about
U turn	.977	.934	Right turn	.954	.925	Right turn	.994	.846	Right turn
Railway crossing	.999	.989	Speed bumps	.975	.925	U turn	.884	.904	Right turn
Two way traffic	.909	.883	U turn	.984	.798	U turn	.944	.957	U turn
Round about	.988	.934	Traffic congestion	.954	.816	Falling rocks	.884	.959	Railway crossing
Road works	.977	.947	Railway crossing	.883	.738	Railway crossing	.843	.889	Speed bumps
Falling rocks	.968	.928	Railway crossing	.835	.746	Road works	.992	.857	Speed bumps
Traffic congestion	.989	.937	Speed bump	.824	.795	Speed bump	.944	.906	Falling rocks
Speed bump	.989	.989	Falling rocks	.813	.794	Falling rocks	.976	.894	Road works
<b>Average</b>	.977	.935		.925	.854		.940	.906	

\*Pre.: Precision.

### 5.1. Discussion

Our proposed flexible DCT grid based technique is accurate in terms of precision and recall in all three combinations of datasets containing images of traffic signs with varying font styles and translated contents. In order to test the performance of our proposed approach under difficult conditions, we tested images of all 20 classes under poor illumination (e.g., images taken in the evening or in bad weather) and partial occlusions (where some other object had blocked a portion of the traffic sign). Table 4 shows the error rate of our proposed traffic sign recognition system under these two conditions. Some traffic signs, e.g. round about and speed limit 100, were recognized with a very small error rate even in poor illumination conditions. Three mandatory signs, i.e. speed limit 90, speed limit 100, and speed limit 60, were recognized with good accuracy under partial occlusion. Overall, the proposed system is less immune to partial occlusions than to poor illumination. This is because the feature set calculated on  $8 \times 8$  flexible grid loses its discrimination ability to a great extent when a portion of the traffic sign is blocked by another object. However, in case of poor lighting conditions, the DCT based feature vectors still possess a reasonably good discrimination ability.

**Table 4.** Error rate of our proposed method under poor illumination and partial occlusion.

Traffic sign	Error rate	
	Poor illumination	Partial occlusion
Speed limit 30	0.234	0.329
Speed limit 40	0.140	0.330
Speed limit 50	0.204	0.244
Speed limit 60	0.093	0.103
Speed limit 70	0.139	0.309
Speed limit 80	0.235	0.296
Speed limit 90	0.304	0.056
Speed limit 100	0.063	0.012
Speed limit 110	0.184	0.222
Speed limit 120	0.240	0.342
Right turn	0.234	0.332
Left turn	0.244	0.343
U turn	0.109	0.303
Railway crossing	0.199	0.109
Two way traffic	0.099	0.304
Round about	0.014	0.234
Road works	0.113	0.330
Falling rocks	0.189	0.430
Traffic congestion	0.200	0.342
Speed bump	0.255	0.233
<b>Average</b>	0.175	0.261

## 6. Conclusion

Generating invariant features from traffic signs with varying writing font styles and sizes is a challenging task. In certain situations, contents of a traffic sign can also be translated into one of the four possible directions. These deviations from the standard can easily fool an automatic traffic sign recognition system. We have presented an effective algorithm to generate invariant features from such non-standard signs. Our proposed algorithm produces promising results for synthetic and a real-world dataset containing more than 1000 images each. We have further shown that our proposed method outperforms two state of the art traffic sign recognition methods for synthetic dataset, real-world dataset, and the combination of the two.

## References

- Fleyeh, H. and Davami, E. "Eigen-based traffic sign recognition", *IET Intelligent Transport Systems*, **5**(3), pp. 190-196 (2011).
- Greenhalgh, J. and Mirmehdi, M. "Real-time detection and recognition of road traffic signs", *IEEE Transactions on Intelligent Transportation Systems*, **13**(4), pp. 1498-1506 (2012).
- Lindner, F., Kressel, U. and Kaelberer, S. "Robust recognition of traffic signals", *IEEE Intelligent Vehicles Symposium*, Parma, Italy, pp. 49-53 (2004).
- Fletcher, L., Apostoloff, N., Petersson, L. and Zelinsky, A. "Vision in and out of vehicles", *IEEE Intelligent Systems*, **18**(3), pp. 12-17 (2003).
- Moutarde, F., Bargeton, A., Herbin, A. and Chanussot, L. "Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular traffic signs recognition system", *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, pp. 1122-1126 (2007).
- Gao, X., Podladchikova, L., Shaposhnikov, D., Hong, K. and Shevtsova, N. "Recognition of traffic signs based on their colour and shape features extracted using human vision models", *Journal of Visual Communication and Image Representation*, **17**(4), pp. 675-685 (2006).
- Escalera, A., Armingol, J. and Mata, M. "Traffic sign recognition and analysis for intelligent vehicles", *Image and Vision Computing*, **21**(3), pp. 247-258 (2003).
- Hsu, S. and Huang, C. "Road sign detection and recognition using matching pursuit method", *Image and Vision Computing*, **19**(3), pp. 119-129 (2001).
- Bascón, S.M., Arroyo, S.L., Jimenez, P.G., Moreno, H.G. and Ferreras, F.L. "Road-sign detection and recognition based on support vector machines", *IEEE Transactions on Intelligent Transportation Systems*, **8**(2), pp. 264-278 (2007).
- Arroyo, S.L., Jimenez, P.G., Bascon, R.M., Ferreras, F.L. and Bascon, S.M. "Traffic sign shape classification evaluation i: Svm using distance to borders", *Proceedings of the IEEE Intelligent Vehicles Symposium*, Las Vegas, USA, pp. 557-562 (2005).
- Barnes, N. and Zelinsky, A. "Real-time radial symmetry for speed sign detection", *Proceedings of the IEEE Intelligent Vehicles Symposium*, Parma, Italy, pp. 566-571 (2004).
- Escalera, A.D.L., Moreno, L.E., Salichs, M.A. and Armingol, J.M. "Road traffic sign detection and classification", *IEEE Transactions on Industrial Electronics*, **44**(6), pp. 848-859 (1997).
- Shadeed, W.G., Nadi, D.I.A. and Mismar, M.J. "Road traffic sign detection in color images", *Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems*, Sharjah, UAE, pp. 890-893 (2003).
- Fang, C.Y., Chen, S.W. and Fuh, C.S. "Road-sign detection and tracking", *IEEE Transactions on Vehicular Technology*, **52**(5), pp. 1329-1341 (2003).
- Wang, Y., Shi, M. and Wu, T. "A method of fast and robust for traffic sign recognition", *Fifth IEEE International Conference on Image and Graphics (ICIG)*, Shanxi, China, pp. 891-895 (2009).
- Gonzalez, R.C. and Woods, R.E., *Digital Image Processing*, 3rd ed., Pearson Education, New Delhi, India, pp. 645-660 (2009).
- Alpaydin, E., *Introduction to Machine Learning*, 1st ed., MIT press, Istanbul, Turkey, pp. 57-103 (2004).
- Fu, M.Y. and Huang, Y.S. "A survey of traffic sign recognition", *IEEE International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Qingdao, China, pp. 119-124 (2010).
- Mogelmose, A., Trivedi, M.M. and Moeslund, T.B. "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey", *IEEE Transactions on Intelligent Transportation Systems*, **13**(4), pp. 1484-1497 (2012).
- Welch, L. "Lower bounds on the maximum cross correlation of signals", *IEEE Transactions on Information Theory*, **20**(3), pp. 397-399 (1974).
- Hoferlin, B. and Zimmermann, K. "Towards reliable traffic sign recognition", *IEEE Intelligent Vehicles Symposium*, Xi'an, China, pp. 324-329 (2009).
- Varan, S., Singh, S., Kunte, R.S., Sudhaker, S.R.D. and Philip, B. "A road traffic signal recognition system based on template matching employing tree classifier", *International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, India, pp. 360-365 (2007).
- Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 886-893 (2005).
- Ghica, D., Lu, S.W. and Yuan, X. "Recognition of traffic signs by artificial neural network", *Proceedings of the IEEE International Conference on Neural Networks*, Perth, WA, Australia, pp. 1444-1449 (1995).

25. Haykin, S.S., *Neural Networks and Learning Machines*, 3rd Ed., pp. 345-420, *Pearson Education, Upper Saddle River*, New Jersey, USA (2009).
26. Ruta, A., Li, Y. and Liu, X. "Real-time traffic sign recognition from video by class-specific discriminative features", *Pattern Recognition*, **43**(1), pp. 416-430 (2010).
27. Carneiro, G., Chan, A.B., Moreno, P.J. and Vasconcelos, N. "Supervised learning of semantic classes for image annotation and retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(3), pp. 394-410 (2007).
28. Mannan, A., Babri, H.A. and Saeed, M. "Offline shape recognition using flexible dct grid", *Scientia Iranica*, **19**(6), pp. 1722-1730 (2012).
29. Oppenheim, A.V. and Schaffer, R.W., *Discrete-Time Signal Processing*, 3rd Ed., pp. 230-259, Prentice-hall, New Delhi, India (2009).
30. Pennebaker, W.B. and Mitchell, J.L., *JPEG: Still Image Data Compression Standard*, 1st Ed., pp. 102-167, Springer Science & Business Media, Springer-Verlag, USA (1993).
31. Schölkopf, B. and Smola, A.J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, 1st Ed., pp. 134-254, MIT press, USA (2002).
32. Canu, S. and Smola, A. "Kernel methods and the exponential family", *Neurocomputing*, **69**(7), pp. 714-720 (2006).
33. Davidson, E.R. "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices", *Journal of Computational Physics*, **17**(1), pp. 87-94 (1975).

## Biographies

**Serosh Karim Noon** completed her MSc degree in Electrical Engineering at University of Engineering and Technology, Lahore, Pakistan, and is presently working towards PhD degree in Electrical Engineering. She is currently serving NFC Institute of Engineering and

Technology, Multan, Pakistan, as a Lecturer in the Department of Electrical Engineering. Her research interests include biomedical image and signal processing, artificial intelligence, and computer vision.

**Kashif Javed** received the BSc, MSc, and PhD degrees in Electrical Engineering in 1999, 2004, and 2012, respectively, from the University of Engineering and Technology (UET), Lahore, Pakistan. He joined the Department of Electrical Engineering at UET in 1999, where he is currently an assistant professor. He has been a reviewer of IEEE Transactions on Knowledge and Data Engineering (TKDE) since 2011. His research interests include machine learning, pattern recognition, and natural language processing for Urdu.

**Abdul Mannan** received BSc and MSc degrees in Electrical Engineering from University of Engineering and Technology, Lahore, Pakistan, and is pursuing PhD degree. Presently, he is serving as assistant professor in the Department of Electrical Engineering at NFC Institute of Engineering and Technology, Multan, Pakistan. His research interests include computer vision, intelligent transportation systems, machine learning, and artificial neural networks.

**Haroon Atique Babri** received the BSc degree in Electrical Engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, in 1981, and the MS and PhD degrees in Electrical Engineering from the University of Pennsylvania in 1991 and 1992, respectively. He was with the Nanyang Technological University, Singapore, from 1992 to 1998, with the Kuwait University from 1998 to 2000, and with the Lahore University of Management Sciences (LUMS) from 2000 to 2004. He is currently a professor of electrical engineering at UET. He has written two book chapters and has more than 60 publications in machine learning, pattern recognition, neural networks, and software reverse engineering.