

Handwritten character recognition using wavelet energy and extreme learning machine

Binu P. Chacko · V. R. Vimal Krishnan ·
G. Raju · P. Babu Anto

Received: 14 August 2010 / Accepted: 5 September 2011 / Published online: 28 September 2011
© Springer-Verlag 2011

Abstract This paper deals with the recognition of handwritten Malayalam character using wavelet energy feature (WEF) and extreme learning machine (ELM). The wavelet energy (WE) is a new and robust parameter, and is derived using wavelet transform. It can reduce the influences of different types of noise at different levels. WEF can reflect the WE distribution of characters in several directions at different scales. To a non oscillating pattern, the amplitudes of wavelet coefficients increase when the scale of wavelet decomposition increase. WE of different decomposition levels have different powers to discriminate the character images. These features constitute patterns of handwritten characters for classification. The traditional learning algorithms of the different classifiers are far slower than required. So we have used an extremely fast leaning algorithm called ELM for single hidden layer feed forward networks (SLFN), which randomly chooses the input weights and analytically determines the output weights of SLFN. This algorithm learns much faster than traditional popular learning algorithms for feed forward neural networks. This feature vector, classifier combination gave good recognition accuracy at level 6 of the wavelet decomposition.

Keywords Character recognition · Feature extraction · Wavelet energy · Extreme learning machine

1 Introduction

Handwriting recognition has been a popular area of research since few decades under the purview of image processing and pattern recognition [1]. A major goal of pattern recognition research is to create human perception capabilities in artificial systems. The task of automatically reading handwriting with close to human performance is still an open problem and the central issue of an active field of research [2]. This may be due to the large degree of variability of human writing. Handwritten character recognition (HCR) systems have to address issues such as infinite variety of character shapes, similarity between characters, and distorted and broken characters [3]. The system having intelligence in recognizing the natural handwriting for all possible scripts around the world is the need of the current era. As a first step of document understanding, a digital image of the document to be analyzed needs to be captured by a scanner or digital camera. Then the segmented characters are subject to a number of preprocessing steps that aim at reducing the variability in the appearance of handwriting [2]. The basic problem is to assign the digitized character to its symbolic class. A pattern recognition algorithm is used to extract shape features and to assign the observed character to the appropriate class. In the case of handprint, it is loosely referred to as intelligent character recognition (ICR) [4]. It provides significant benefit to man–machine communication.

The techniques for automatic handwriting recognition can be distinguished as being either online or offline depending on the particular processing strategy applied. Online recognition is performed as the text to be recognized is written. In contrast, offline recognition is performed after the text has been written [2]. It deals with the analysis of spatio-luminance of an image. Two critical

B. P. Chacko (✉) · V. R. Vimal Krishnan · G. Raju ·
P. Babu Anto
Department of Information Technology,
Kannur University, Kannur, India
e-mail: binupchacko@gmail.com

issues of developing a handwriting recognition system are the selection of a feature set and designing a classifier. The state of the art classifiers include statistical classifier such as modified quadratic discriminant function (MQDF), neural classifiers such as multi layer perceptron (MLP), radial basis function (RBF) classifier, polynomial classifier (PC), learning vector quantization (LVQ) and support vector machine (SVM) [5]. The state of the art features in HCR are the different variants of direction features [6]. These include chain code [7], gradient [8], and curvature features [9]. Gabor transform [10] and statistical/structural features [11] have also been successfully used in character recognition research. Shi et al. [9] considered a composite feature by combining gradient and curvature feature vectors.

The aim of feature extraction of the recognition structure is to extract discriminant information from an image of a character as well as to reduce its dimension of representation. This reduction is required to make the conception of the classification system easier, when the discriminant feature extraction allows presenting competently a character to the classifier. In this paper, we adapt a feature called wavelet energy (WE) computed from wavelet transform (WT). Wavelets are powerful tools of multi-resolution analysis, which have been widely used in image processing applications. A 2D wavelet transform can decompose the image in several directions at different scale. Moreover, to a non oscillating pattern, the amplitudes of wavelet coefficients increase when the scale of wavelet decomposition increase. WE of different decomposition levels have different powers to discriminate character images [12]. Computation of WE parameter is easy, and it produces better recognition accuracy. The main advantage of using this parameter is its ability to separate the character from noise in the wavelet transformed domain [13].

Once discriminant features have been extracted, they are submitted to a logic decision system whose task is to identify the character that they represent. Over the years, artificial neural networks (ANN) have widely used in classification tasks. The popularity of neural networks is due to their ability to approximate complex nonlinear mappings directly from the input samples [14]. However, their performance is strongly affected by the quality of the representation of the characters. This may require a large number of parameters to represent the character. Moreover, the gradient based learning algorithms (back propagation) may easily get stuck in a local minimum. Also, the activation functions need to be differentiable [15]. The computation time will also increase with the increase of network size. So, it is necessary to perform efficient features extraction on the one hand, and to take steps to reduce the training/testing time on the other hand. ELM is an

efficient algorithm which tends to reach the smallest training error, obtain smallest norm of weights, produce best generalization performance, and runs extremely fast [16].

The remaining part of this paper is organized as follows. The next section reviews the work done on HCR for various language domains. Section 3 introduces the derivation and analysis of the WE parameter. ELM is briefly explained in Sect. 4. An experiment on handwritten Malayalam character recognition using wavelet energy feature (WEF) and ELM is given in Sect. 5. This paper concludes in Sect. 6 by specifying the advantages of using this approach for HCR.

2 Previous work

The constant development of computer tools leads to the requirement of easier interfaces between man and computer. Such an interface can be created with offline systems which have significant economic impact on specialized domains such as interpreting handwritten postal addresses on envelopes, automatic processing of bank cheques and official forms, offline signature verification, whiteboard reading, etc. [2]. The importance of these applications has led to intense research for several years in the field of offline HCR.

2.1 Handwritten character recognition

A general framework of handwriting recognition involves tasks such as preprocessing (noise removal, text detection or similar), segmentation, representation (skeleton, contour, pixels or other), feature extraction and recognition. Some approaches do not use all these elements but only a subset [17]. One may find so much of work for languages like Arabic, Chinese, and Indian scripts Hindi, Kannada, Tamil, Bengali, Malayalam, and Gujarati. Offline HCR systems are matured only in few languages like English and Chinese [18]. The nature of character set (language) has great influence on the performance of HCR system.

Abuhaiba et al. [19] proposed a set of character graph models to recognize isolated Arabic letters. Each model was a state machine with transitions corresponding to directions of segments in the character and with additional fuzzy constraints to distinguish some characters. Amin et al. [20] also used a skeleton based graph representation for the recognition of Arabic letters. Structural features including curves were fed into a five layer neural network. Both image level [21] and structural features [22] have been applied to Arabic handwriting recognition. The former places the burden of processing on the recognizer, while the latter involves more processing at the feature

detection stage. More training data would be needed for image level features to model handwritten character shapes. Structural and hybrid approaches are more common in handwritten Arabic. Al-Shaher and Hancock considered the recognition problem from a different perspective [23]. They choose seven basic shape classes found in Arabic characters, each of which consisted of only one trajectory. Their system distributed twenty points uniformly along each trajectory to train point distribution models (PDMs). Test result showed that a mixture of PDMs achieved significantly better performance than did a single PDM. The uncommon representation of contour projections is employed by Dehghani et al. [24] for Persian character recognition. The image is projected in multiple directions and the chain code of the contour of each projection is obtained. The contour is sampled, and the features are obtained for each section using a two dimensional pattern, the number of active pixels, and slope and curvature. Separate feature vectors from the contours of horizontal and vertical projections are computed and modeled by individual hidden Markov models (HMM), yielding two HMMs per character. During recognition, the scores from individual classifiers are integrated to improve the performance.

HMMs are also appropriate for learning characteristics that are difficult to describe intuitively. Today, systems based on Markov models (MM) are successfully used for automatic handwriting recognition. In [25], HMMs are used for the recognition of both legal and courtesy amounts in bank cheques. A cheque reading system that recognizes both legal and courtesy amounts on French cheques in real time with a read rate of 75% and an error rate of 1 in 10,000 is described in [26]. It uses a segment and recognize (using multiple classifiers) paradigm. The approach of first recognizing the legal amount to drive the recognition of the courtesy amount is used in [27]. Albert et al. [28] constructed leave-one-out-training and leave-one-out-testing HMMs (LOOT-HMM), and carried out an experiment on handwritten digit recognition problem. In a sequence of M observations, leave an observation out of the original sequence and make a new sequence without this observation. Repeat the process on all observations in this sequence and create M new sequences. Then, sum up the likelihood of all M sequences to make a final decision.

Obtaining high recognition accuracy is often difficult for a single classifier due to the enormous variability in the handwritten shapes. One solution is to employ a cascade of classifiers with a rejection strategy. Nunes et al. [29] used a cascade of classifiers along with a recognition scheme having low computational complexity. Giusti et al. [30] presented a two stage recognition system in which the patterns rejected by the first classifier were classified by the second classifier. Cao et al. [31] used a multistage classifier

system comprising two different neural networks for the recognition of handwritten numerals in two stages. Georgios et al. [32] presented a feature extraction technique based on recursive sub division of the character image so that the resulting sub images contain balanced number of foreground pixels. Feature extraction is followed by a two stage classification scheme based on the level of granularity of the feature extraction method. Classes with high values in the confusion matrix are merged at a certain level, and for each group of merged classes, granularity features from the level that best distinguishes them are employed.

Lei et al. [33] proposes an optimal dimensionality reduction method based on the integration of geodesic paths and non-parametrical dimensionality reduction. In order to solve large scale pattern recognition problems like Chinese character recognition, the authors present a simplification strategy for algorithms. This approach greatly speeds up training by employing non-parametrical dimensionality reduction algorithms optimized with geodesic distance. It is more effective than principle component analysis (PCA) and linear discriminant analysis (LDA) with stronger classification ability. Tsukumo and Tanaka [34] proposed 1D nonlinear normalization methods which normalize an input character into a relatively stable shape and successfully applied to character recognition. Fu and Xu [35] applied Bayesian decision based neural networks for classification, whilst Mao and Mohiuddin [36] introduced character degradation models for training the ensemble of character classifiers using a boosting algorithm.

Several experiments have been conducted using the database ETL-9B. Dong et al. [37] has achieved a recognition rate of 99% using SVM. Liu et al. [38] proposed 1D moment and bi-moment normalization methods using quadratic curve fitting. They also proposed 2D normalization methods [39] to reduce the runtime complexity, and obtained a recognition rate of 99.19%. Liu et al. [40] proposed MQDF3, and discriminately trained the parameters to boost the recognition rate to 99.39%. Kato et al. [41] employed slant correction and an asymmetric Mahalanobis distance for classification, and attained a recognition rate of 99.42%. One problem in the training of statistical classifiers is the lack of sufficient training samples. The problem can be alleviated by reducing the feature dimension using Fisher's discriminant, as well as by regularizing the covariance matrix as is done in MQDFs. A third technique is to augment the training set by generating a huge number of artificial distorted samples from the existing training samples. Recently, Leung et al. [42] combined all the three techniques and boosted the recognition rate to 99.46%. Leung and Leung [43] propose critical region analysis technique to tackle the problem of similar character classes. The central idea of the critical region analysis method is to highlight the particular cells (critical regions) that can

distinguish between two similar character classes by extracting finer features from these regions. The extra features extracted are appended to the original feature vector so that the overall contribution of the critical regions in the feature vector is increased. The authors adopted a hierarchical multi classifier approach, in which a two class classifier is used for each pair of similar character classes called confusing pair. The input character is first classified by the first level classifier. If it is identified as belonging to a confusing pair, the corresponding two class classifier is invoked to give a final classification; otherwise, the decision from the first classifier is final. This approach produced the highest recognition accuracy of 99.53%.

Pattern recognition studies on various Indian scripts are also widely reported. Chinnuswami et al. [44] presented their work to recognize handwritten Tamil characters. In this work, authors have used the structure of Tamil characters. Using the curves and strokes of characters, features were identified and statistical approach was used for the classification. Shanthi et al. [45] used SVM for the recognition of handwritten Tamil characters. In this work image subdivision was used for feature extraction. Sukhaswami et al. [46] used multiple neural network associate memory for the recognition of both printed and handwritten Telugu characters. Many preprocessing activities such as smoothing, thinning, skew detection and correction, and normalization can be found in the work done on Gujarati numeral recognition [47]. A skew angle detection of Indian script documents is proposed in [48]. Rajasekhara Aradhya [49] proposed an offline handwritten numeral recognition system for four south Indian languages Kannada, Tamil, Telugu and Malayalam. In this work, author suggested a feature extraction technique based on zone and image centroid. The author has used two different classifiers, nearest neighbourhood and back propagation neural network, to achieve 99% accuracy for Kannada and Telugu, 96% for Tamil and 95% for Malayalam.

Dutta et al. [50] recognized both printed and handwritten alphanumeric Bengali characters using curvature features. Here, features like curvature maxima, curvature minima, and inflexion points were considered. Thinning and smoothing were also performed prior to classification of characters. Bhattacharya et al. [51] described a hybrid scheme, while Bhowmick et al. [52] used stroke features for the recognition of handwritten Bengla numerals and characters, respectively. Ujjwal and Chaudhuri [5] presented a multistage cascaded recognition scheme using wavelet based multi resolution representations and MLP classifiers for the recognition of mixed handwritten numerals Bengla, Devanagari and English. In this scheme, a numeral is subjected to three MLP classifiers corresponding to three coarse to fine resolution levels in a cascaded manner. If rejection occurs even at the highest

resolution, another MLP is used as the final attempt to recognize the input numeral by combining the outputs of three classifiers in previous stages. Hanmandlu and Murthy [53] proposed a fuzzy model based scheme for the recognition of handwritten Devanagari numerals. Banashree et al. [54] attempted the identification handwritten Hindi digits, using diffusion half toning algorithm. Here, a sixteen segment display concept has been used for feature extraction.

Lajish [55] has applied several techniques for the recognition of handwritten Malayalam characters. The author has used features like state space point distribution, zoned vector distance and fuzzy zoned normalized vector distance (FZ-NVD), and the classifiers c-means clustering, k-NN, class modular neural network and adaptive neuro fuzzy inference system (ANFIS). The highest recognition rate of 82.35% is obtained for FZ-NVD-ANFIS combination. This work pointed out the need of other discriminating features for this pattern recognition problem. Bindu and Raju have done some works on this problem using MLP and NVD [56–60]. The authors have explored the meshing techniques to divide the thinned image into 16 blocks and measured NVD from it. In [60], a recognition accuracy of 76.21 and 76.45% is obtained when NVD is measured from common origin (global elastic meshing) and local origin (fuzzy boundary based meshing), respectively. When the local elastic meshing with common origin is used, the recognition rate is increased to 79.8% [56]. An improved recognition accuracy of 83.31% is obtained by measuring the NVD from local centroid (fixed meshing) [57]. The accuracy is increased to 88.22% by using three more features namely, aspect ratio, image code, and the distance between image origin and centroid. In [58], multiple MLP classifiers are used to boost the recognition rate to 93.3%. A QDF based character recognition system attained an accuracy of 89.99%, in which each thinned image is divided into 64 blocks [59]. In all these works, the poor performance of thinning algorithm affected in the recognition accuracy. Binu and Babu [61] used discrete features and MLP to achieve a recognition rate of 90.18%. In this work, efficient skeletonization and skeleton pruning algorithms are used to get better features. Still, the parasitic components created due to writing style of the people are unavoidable. Wavelet based feature extraction method is also used in Malayalam character recognition [3, 62, 63]. Binu et al. [62] compared the performance of back propagation (BP) and multiple back propagation (MBP) algorithms using wavelet mean based features. The work on binary images showed the superiority of MBP with a recognition rate of 80.85%. Raju [63] investigated the possibility of using 1D wavelet transform of vertical and horizontal projection profiles of thinned characters. This work is extended in [3] by taking the average of wavelet

coefficients as the feature. When the aspect ratio is also included in the feature set, the recognition rate is increased to 81.3% using MLP. Even though wavelet mean is used in different environments, the recognition rate does not go beyond a certain limit. This clearly shows the lack of discriminating power of this feature.

The above HCR systems have proposed many feature extraction methods and classification techniques for different languages. The approaches found suitable for one language need not give the same level of performance for another language. Even though different combination of features and classifiers were used in Malayalam character recognition, a remarkable achievement is not yet obtained. The experimental result showed that the classifiers were not efficient w.r.t. time either in training stage or in testing stage. The different feature sets also could not contribute much to improve the recognition accuracy. This really motivated us to search for a new set of features and an efficient classifier. In this work, we have used ELM, a fast learning algorithm, and a wavelet based feature set known as wavelet energy. ELM is faster than SVM and has better generalization ability [64]. It gave better recognition accuracy than SVM, HMM, and nearest mean classifiers [65]. Some works based on wavelet features are given in the next section. The discriminating power of these features mainly depends on the choice of wavelet, decomposition level, and further processing of wavelet coefficients. Here, the WEF is measured as the square of wavelet coefficients, and the suitable wavelet and decomposition level are determined through experimentation. Different from the previous methods of extracting features from thinned images, this approach takes features from binary images.

3 Wavelet transform

Wavelets are localized basis functions which are translated and dilated versions of some fixed mother wavelet. They form a foundation for representing signals, such as images, in a hierarchy of increasing resolutions. This multi resolution analysis enables us to analyse the image in different frequency bands. WT is a technique for analyzing the time–frequency domain that is most suited for a non stationary signal. It uses the localized basis function to capture the localized features of a signal. Hence, it provides better signal approximation than other tools such as Fourier, sine or cosine transform [13]. The ability to capture local information is essential because characters are locally quite different [66]. Wavelet analysis provides immediate access to information that can be obscured by other time–frequency methods such as Fourier analysis.

The application of WT to a wide variety of problems is so plentiful that they have emerged as the most promising

techniques in the past decade. Wavelets helped to analyse variations of values at financial markets. Biologists used them for cell membrane recognition; the federal bureau of investigation considered the wavelet application for storage of 30 million sets of criminal fingerprints; computer scientists exploited them in image processing like edge recognition, image searching, animation control, image compression and even Internet traffic description; engineers used WT for time phenomenon study in transient process. Recently, wavelets have been implemented for structural health monitoring and damage detection. In seismic engineering, wavelets analyse seismic records for geologic purposes [67].

The WT decomposes the original image into four sub bands: approximation and the three detail sub bands in the horizontal, vertical and diagonal directions (Fig. 1). This process can be iterated until a predefined number of levels to obtain the multi resolution representation of the image. The levels refer to the inverse of the size of aperture of the wavelet filter to be applied on the image. As the WT requires downsampling by two from one level to another, the maximum level of applying the WT depends on the number of data points in the data set. For an image with a size of s times s pixels, the number of levels is M , where $2^M = s/2$ [68].

3.1 Feature extraction using wavelets

Features are information passed to the recognizer, such as pixels, shape data, or mathematical properties [17]. Some papers have been published using WT as a feature extractor. In the last 20 years, wavelet and similar transformations became popular for image processing tasks including character recognition. In image processing, discrete wavelets with finite support and compact are more

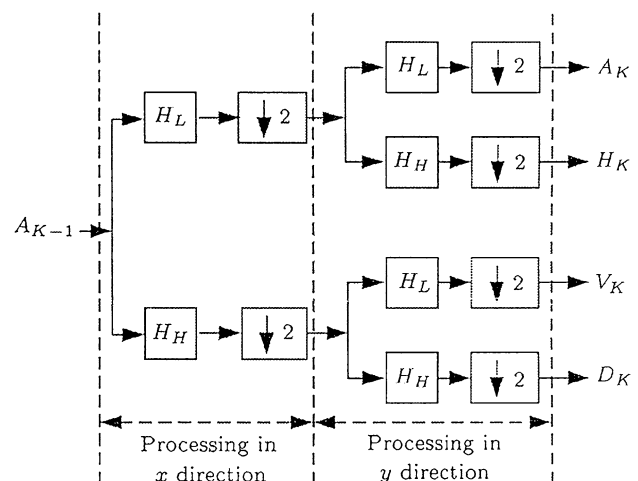


Fig. 1 2D discrete wavelet transform [12]

popular due to their relations to multi-resolution filter banks. These wavelets have finite impulse response [3]. Use of multi resolution analysis for character recognition is found in [69] and [70]. Favata et al. [69] implemented a multi resolution approach to feature generation by computing the features at three stages: local, global and intermediate. The dynamic character recognizer proposed by Park et al. [70] starts with features extracted at a coarse resolution while finer resolution of a sub image is considered on each successive recursive passes until the classification meets certain acceptance criteria. In [5], the wavelet filter is applied three times, and the chain code histogram feature is extracted from each of the detailed image components $D_{LL}^{(k)}$, $k = 1, 2, 3$. For computation of features from $D_{LH}^{(k)}$, $D_{HL}^{(k)}$, and $D_{HH}^{(k)}$, the bounding box of each of them is divided into 8×8 blocks and the number of black pixels is counted. The feature vector is formed by concatenating $D_{LL}^{(k)}$, $D_{LH}^{(k)}$, $D_{HL}^{(k)}$, and $D_{HH}^{(k)}$.

The zero crossings of the wavelet transform provide the locations of sharp signal variation. [71] used the count of zero crossings at different scale as a feature. In [72], feature extraction is done by taking the wavelet packet transform of the character image (using best basis algorithm) for a desired number of multi resolution levels. In [73], the authors proposed a scheme to recognize unconstrained handwritten numerals using Haar wavelets to extract multi resolution features. Two types of feature vectors were considered. The first type used only the features at one resolution level and the second type used all the features at two resolution levels. [74] proposed a method to recognize unconstrained handwritten numerals using spline wavelet CDF 3/7. In this system, bi-dimensional wavelet transform was applied and the four resulting sub band images of coefficients were used as feature vector. Minhua et al. [66] decomposed the character skeleton at two scales of resolution and yielded seven sub band images. Each sub band image is further divided into sub blocks and extracted WE. The sum of wavelet coefficients in the sub block/sub band image is taken as the WE. Wavelet energy density (WED) of a sub block is calculated by dividing its WE with that of the corresponding sub band image. The WEDs of all the sub blocks together constitute the WED feature.

3.2 Wavelet energy

The features extracted from wavelet coefficients are numerous, including total energy, statistical representation such as mean and standard deviation, histogram properties and co-occurrence matrix features. A simple and powerful feature extracted from the wavelet coefficients is the WE of approximation and detail images. Such energy signatures provide a good indication of the total energy contained at specific spatial frequency levels and orientations. WEF

computed using Discrete Wavelet Transform (DWT) is one of the best features to be used in the classification process. It is also robust to some extent of rotation and translation of the images. This paper also analyses the discriminabilities of different wavelets at each level. The experimental result shows that the order of discriminability increases as the level increases.

In order to extract the features, WT is applied to each binary image. The wavelet coefficients from the approximation sub band and the detail sub bands of all the decomposition levels are used to formulate the WEF [68]. It is determined using the following equations.

WE in horizontal, vertical and diagonal directions at i th level is defined as:

$$E_i^h = \sum_{x=1}^P \sum_{y=1}^Q (H_i(x, y))^2 \quad (1)$$

$$E_i^v = \sum_{x=1}^P \sum_{y=1}^Q (V_i(x, y))^2 \quad (2)$$

$$E_i^d = \sum_{x=1}^P \sum_{y=1}^Q (D_i(x, y))^2 \quad (3)$$

These energies reflect the strength of the images' details in different directions at the i th wavelet decompose level. The energy of non oscillating patterns is concentrated at the large wavelet decomposition scales. The feature vector,

$$(E_i^h, E_i^v, E_i^d)_{i=1,2,\dots,M} \quad (4)$$

where M is the total wavelet decomposition level, can describe the global details information of a character image efficiently. Since the energy values of components obtained from decomposed images varies widely, the normalized energy values are used to develop our recognition method. Normalization is performed by the total energy calculated using Eq. (5). This normalized vector is called the WEF. WEF is composed of different level WEFs, which have different abilities to discriminate handwritten characters [12].

$$E_{total} = \sum_{k=1}^K \sum_{i=1}^R (c_i^k)^2 \quad (5)$$

where R is the total number of wavelet coefficients of each sub band, K is the total number of sub bands, and c_i^k is the i th coefficient of the k th sub band.

WE of the approximation sub band can be similarly determined using approximation coefficients.

4 Single hidden layer feed forward network

The research on approximation capabilities of feed forward neural networks has focused on two aspects: universal

approximation on compact input sets and approximation in a finite set of training samples. In real applications, neural networks are trained in finite training set. For function approximation in a finite training set, Huang and Babri [75] show that a single hidden layer feed forward network (SLFN) with at most N hidden nodes and with almost any nonlinear activation function can exactly learn N distinct observations. According to conventional neural network theories, SLFN with additive or RBF hidden nodes are universal approximators when all the parameters of the networks are allowed adjustable. However, as observed in most neural network implementations, tuning all the parameters of the networks may cause learning complicated and inefficient, and it may be difficult to train networks with non-differentiable activation functions such as threshold networks. Huang et al. [16] proved that the input weights and hidden layer biases of SLFN can be randomly assigned if the activation functions in the hidden layer are infinitely differentiable. In order to let SLFNs work as universal approximators, the network sequence $\{f_{\hat{N}}\}$ with randomly generated hidden nodes can coverage to any continuous target function f by only adjusting the output weights.

Let $L^2(X)$ be a space of functions f on a compact subset X in the d -dimensional Euclidean space \mathbf{R}^d such that $|f|^2$ are integrable, i.e., $\int_X |f(x)|^2 dx < \infty$. Let $e_{\hat{N}} \equiv f - f_{\hat{N}}$ denote the residual error function for the current network $f_{\hat{N}}$ with \hat{N} hidden nodes. Huang et al. [14] proved that given any bounded non-constant piecewise continuous activation function $g: \mathbf{R} \rightarrow \mathbf{R}$ for additive nodes or integrable piecewise continuous activation function $g: \mathbf{R} \rightarrow \mathbf{R}$ (and $\int_{\mathbf{R}} g(x) dx \neq 0$) for RBF nodes, for any continuous target function f and any randomly generated sequence $\{g_{\hat{N}}\}$, $\lim_{\hat{N} \rightarrow \infty} \|e_{\hat{N}}\| = 0$ holds if $\beta_{\hat{N}} = \langle e_{\hat{N}}^{-1}, g_{\hat{N}} \rangle / \|g_{\hat{N}}\|^2$. The network with fixed architecture and randomly assigned hidden nodes is called ELM where the output parameters are determined by ordinary least square and according to $\lim_{\hat{N} \rightarrow \infty} \|f - f_{\hat{N}}\| = 0$ [14].

4.1 Extreme learning machine

The slow learning speed of feed forward neural networks has been a major bottleneck in their applications during the past decades. The main reason is that all the parameters of the networks are tuned iteratively when the gradient-based learning algorithms are used for training. Unlike these conventional implementations, this paper describes a learning algorithm called ELM for SLFNs. In the application of SLFN, input weights and hidden layer biases are randomly chosen, and they need not be adjusted at all. This method not only makes the learning extremely fast but also produces good generalization performance. After

parameters are chosen randomly, SLFN can be simply considered as a linear system and the output weights (linking the hidden layer to the output layer) of SLFN can be analytically determined through simple generalized inverse operation of the hidden layer output matrices. Based on this concept, this section explores ELM whose learning speed can be thousands of times faster than traditional feed forward neural network learning algorithms like back propagation.

For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbf{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbf{R}^m$, standard SLFN with \hat{N} hidden neurons and activation function $g(x)$ are mathematically modeled as

$$\sum_{i=1}^{\hat{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, \quad j = 1, 2, \dots, N \tag{6}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ is the weight vector connecting the i th hidden neuron and the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is the weight vector connecting the i th hidden neuron and the output neurons, and b_i is the threshold of the i th hidden neuron. The number of input and output neurons is represented using n and m respectively. $\mathbf{w}_i \cdot \mathbf{x}_j$ denotes the inner product of \mathbf{w}_i and \mathbf{x}_j . The output neurons are chosen linear in this experiment.

The SLFN with \hat{N} hidden neurons with activation function $g(x)$ can approximate these N samples with zero error means that $\sum_{j=1}^N \|\mathbf{o}_j - \mathbf{t}_j\| = 0$, i.e., there exist β_i, \mathbf{w}_i and b_i such that

$$\sum_{i=1}^{\hat{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \quad j = 1, 2, \dots, N \tag{7}$$

The above N equations can be written compactly as:

$$\mathbf{H}\beta = \mathbf{T} \tag{8}$$

where

$$H(\mathbf{w}_1, \dots, \mathbf{w}_{\hat{N}}, b_1, \dots, b_{\hat{N}}, \mathbf{x}_1, \dots, \mathbf{x}_N) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & g(\mathbf{w}_{\hat{N}} \cdot \mathbf{x}_1 + b_{\hat{N}}) \\ \vdots & & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \dots & g(\mathbf{w}_{\hat{N}} \cdot \mathbf{x}_N + b_{\hat{N}}) \end{bmatrix}_{N \times \hat{N}} \tag{9}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\hat{N}}^T \end{bmatrix}_{\hat{N} \times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m} \tag{10}$$

\mathbf{H} is called the hidden layer output matrix of the neural network; the i th column of \mathbf{H} is the i th hidden neuron's output vector with respect to inputs $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ [16].

If the number of hidden neurons is equal to the number of distinct training samples, $\hat{N} = N$, matrix \mathbf{H} is square and invertible, and SLFN can approximate these training

samples with zero error. However, in most cases the number of hidden neurons is much less than the number of distinct training samples, $\hat{N} \ll N$, \mathbf{H} is a non square matrix and there may not exist $\mathbf{w}_i, b_i, \beta_i$ ($i = 1, \dots, \hat{N}$) such that $\mathbf{H}\beta = \mathbf{T}$ [16]. Thus, instead one may need to find specific $\hat{\mathbf{w}}_i, \hat{b}_i, \hat{\beta}$ ($i = 1, \dots, \hat{N}$) such that

$$\begin{aligned} & \| \mathbf{H}(\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_{\hat{N}}, \hat{b}_1, \dots, \hat{b}_{\hat{N}})\hat{\beta} - \mathbf{T} \| \\ &= \min_{\hat{\mathbf{w}}_i, \hat{b}_i, \hat{\beta}} \| \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_{\hat{N}}, b_1, \dots, b_{\hat{N}})\beta - \mathbf{T} \| \end{aligned} \quad (11)$$

The hidden layer output matrix \mathbf{H} can remain unchanged once random values have been assigned to the input weights \mathbf{w}_i and the hidden layer biases b_i at the beginning of learning. The training of an SLFN is simply equivalent to finding a least square solution $\hat{\beta}$ of the linear system $\mathbf{H}\beta = \mathbf{T}$:

$$\begin{aligned} & \| \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_{\hat{N}}, b_1, \dots, b_{\hat{N}})\hat{\beta} - \mathbf{T} \| \\ &= \min_{\hat{\beta}} \| \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_{\hat{N}}, b_1, \dots, b_{\hat{N}})\beta - \mathbf{T} \| \end{aligned} \quad (12)$$

According to theorem 3.1, the smallest norm least squares solution of the above linear system is:

$$\hat{\beta} = \mathbf{H}^\dagger \mathbf{T} \quad (13)$$

Theorem 3.1: *Let there exist a matrix \mathbf{G} such that $\mathbf{G}\mathbf{y}$ is a minimum norm least squares solution of a linear system $\mathbf{A}\mathbf{x} = \mathbf{y}$. Then, it is necessary and sufficient that $\mathbf{G} = \mathbf{A}^\dagger$, the Moore–Penrose (MP) generalized inverse of matrix \mathbf{A} [16].*

Algorithm ELM: Given a training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N\}$, activation function $g(x)$, and hidden neuron number \hat{N} ,

Step 1: Assign arbitrary input weight \mathbf{w}_i and bias b_i , $i = 1, \dots, \hat{N}$.

Step 2: Calculate the hidden layer output matrix \mathbf{H} .

Step 3: Calculate the output weight β :

$$\beta = \mathbf{H}^\dagger \mathbf{T} \quad (14)$$

where \mathbf{H} , β and \mathbf{T} are defined as Eqs. (9) and (10).

The MP generalized inverse and minimum norm least squares solution of a generalized linear system play an important role in the development of ELM algorithm. The learning time of ELM is mainly spent on calculating MP generalized inverse of the hidden layer output matrix \mathbf{H} . The activation functions used are differentiable and non differentiable, and continuous and non continuous. Such functions include sigmoidal, radial basis, sine, cosine, exponential, etc. [16]. ELM avoids many difficulties faced by gradient based learning methods such as stopping criteria, learning rate, epochs, and local minima [15]. Also, it replaces the tuning of hidden node parameters by random assignment, and then problem is solved by computing a

simple least squares solution rather than a time consuming optimization.

4.2 Types of ELM

During the past five years, many variants of ELM have been proposed. The online sequential mode of ELM is given in [76]. This algorithm is referred to as online sequential ELM (OS-ELM), which can learn data one-by-one or chunk-by-chunk with fixed or varying chunk size. Only block size and number of hidden nodes are used as control parameters. OS-ELM is faster than other sequential learning algorithms and produces better generalization performance. Usually, ELM may need higher number of hidden neurons due to the random determination of the input weights and hidden biases. In [15], a modified differential evolution (DE) algorithm is used to search for the optimal input weights and hidden biases. This approach is known as evolutionary ELM (E-ELM). Evolutionary algorithms (EA) are widely used as a global searching method for optimization. E-ELM is able to achieve good generalization performance with much more compact networks.

The generalization performance of ELM for sparse data classification problem depends on three parameters—number of hidden neurons, input weights, and bias values. In [77], the authors proposed two approaches, real coded genetic algorithm based ELM (RCGA-ELM) and sparse ELM (S-ELM), to set these parameters. RCGA-ELM used two genetic operators for this purpose. The network based genetic operator controls the number hidden neurons and the weight based genetic operator evolves the input weight and bias values. S-ELM (KS-ELM [78]), which is computationally less intensive than RCGA-ELM, uses a K-fold cross-validation (CV) to select the ELM parameters. CV approach is used to select the input weights and bias values for the given number of hidden nodes such that the estimated generalization error is small. Then, this process is repeated with a random set of hidden nodes and determined the optimal number of hidden nodes for which training and CV efficiencies are high.

A number of algorithms have been proposed to choose the network structure of different applications. They are categorized into pruning method, forward selection method, and incremental learning algorithms. Rong et al. [79] has proposed a pruned ELM (P-ELM) for pattern classification applications, which starts with a large network and then eliminates the hidden nodes with low relevance to the class label. The statistical criteria such as Chi-square (χ^2) and information gain are utilized to check the relevance. Miche et al. [80] presented optimally pruned ELM (OP-ELM), in which hidden nodes are ranked by multi response sparse regression algorithm. The final model is selected by leave-one-out CV. A constructive hidden nodes selection method

for ELM (CS-ELM) is proposed in [81] based on a stepwise forward selection method. At each step, the hidden node with an output that has the highest correlation with the current residual is selected. The hidden nodes selection stops automatically when the unbiased risk estimation criterion reaches its minimum. In [82], Lan et al. has proposed a two stage ELM (TS-ELM) consisting of forward selection stage and backward elimination stage. In forward selection, many groups of hidden nodes are randomly generated in each step, and the group of hidden nodes with the highest net contribution is selected and added to the network. The selection process is terminated when the stopping criterion of final prediction error achieves its minimum. In backward elimination, the hidden nodes selected in the previous stage are reviewed one by one, and insignificant hidden nodes are removed from the network.

In [14], Huang et al. developed an incremental ELM (I-ELM), in which nodes are added one by one to the hidden layer, while keeping the output weights of the existing hidden nodes unchanged. I-ELM requires only two control parameters—target error and maximum number of hidden nodes. In [83], Barron's convex optimization learning method [84] is incorporated into I-ELM, and the resulting algorithm is referred to as convex I-ELM (CI-ELM). Different from I-ELM, CI-ELM recalculates the output weights of the existing hidden nodes after a new hidden node is added. It can achieve faster convergence rate, more compact network architecture, better generalization performance, and more stability than I-ELM.

It is found that some of the hidden nodes in I-ELM may play a very minor role in the network output and this may eventually increase the network complexity. In order to avoid this issue and to obtain more compact network architecture, Huang and Lei [85] proposed an enhanced method for I-ELM (EI-ELM). At each learning step, several hidden nodes (say k nodes, where k is a constant) are randomly generated and among them the hidden node leading to smallest residual error will be added to the existing network. EI-ELM can achieve faster convergence rate and much more compact network architecture than I-ELM. I-ELM is a specific case of EI-ELM when $k = 1$. Guorui et al. [86] proposed a method to automatically determine the number of hidden nodes in generalized SLFN. This approach is referred to as error minimized ELM (EM-ELM), and can add hidden nodes one by one or group by group. If the network output error is less than the target error, hidden nodes will not be added and the learning terminates. Otherwise, hidden nodes are added and generate new hidden layer output matrix. EM-ELM reduces the computation complexity by updating only the output weights incrementally. It is much faster than other sequential/incremental/growing algorithms, and produce good generalization performance.

5 Experimental result

Experiments have been conducted to extract the features and thereby classify the Malayalam characters. One of the most popular and efficient wavelet features is the energy of wavelet decomposed images. WE reflect the distribution of energy along the frequency axis over scale and orientation, and have been proven to be very powerful for pattern classification. Over the last 20 years, numerous wavelets have been proposed which possess different properties, and are optimal or near optimal in a particular sense. The experiments are conducted with different wavelets, and we obtained good results with daubechies, symlet and biorthogonal wavelets. The DWT, based on sub band coding, is found to yield a fast computation of WT. It is easy to implement and reduces the computation time and resources required. The DWT is applied in size normalized binary images. After the M th level wavelet transform, the wavelet energy for each sub band is computed in order to form the feature vector. Two different feature sets are created, the first one contains only detail sub bands, and the other includes approximation sub band also. It is found that the first set could give 3% more recognition accuracy than the second set, and so only the detail sub bands are used in this classification.

The different variants of ELM are either batch learning or sequential learning. These algorithms determine SLFN parameters randomly or experimentally. For the purpose of comparison, an algorithm from each group is used for experimentation. The recognition of handwritten characters is performed with ELM, E-ELM (batch learning and parameters are determined experimentally), and OS-ELM (sequential learning and parameters chosen randomly). All the experiments are implemented in MATLAB 7.10 environment using a PC with Core 2 Duo 2.0 GHz processor and 3 GB RAM. This process involves both training stage and recognition stage. The collected samples are divided randomly into training set and testing set in the ratio 2:1. In this work, a total of 9,000 characters of 30 different classes collected from 300 persons are used. The sample images of these characters are shown in Fig. 2. In the training stage, WEFs of the training samples are used. In the recognition stage, WEFs of the testing samples are used. The input layer of SLFN contains 3 M nodes to accommodate the feature vector. The output layer contains 30 nodes, equivalent to the number of classes. Generally, the number of nodes in the hidden layer is decided by a rule of thumb, and it is different for different learning algorithms. The sine function is used as the activation function for this character recognition problem.

The experiments with E-ELM gave a wide variability in classification results with respect to the feature sets. It is found that the number of hidden neurons is different for

Fig. 2 Malayalam character set



different levels of wavelet decomposition. As the decomposition level increases, the feature size will also increase. This will necessitate more memory for feature set, which in turn decrease the number of hidden neurons. In this pattern classification problem, E-ELM supported 650, 800, 950 and 1250 hidden neurons at the decomposition levels 6, 5, 4 and 3 respectively. Beyond this limit, the algorithm ran out of memory. The lack of hidden nodes affected the performance of E-ELM. The recognition rate varies from 66.52% (bior1.3 at level 6) to 76.84% (db5 at level 6) in different feature sets (Table 1). The training time ranges from 112 s (bior1.5 at level 6) to 502 s (db2 at level 3), whereas testing consumed 0.17 s (sym3 at level 6) to 0.31 s (db2 at level 3). The feature set based on db5 at sixth decomposition level produced highest recognition accuracy of 76.84% by spending 114 s for training and 0.27 s for testing.

In many applications, data may arrive one-by-one or chunk-by-chunk during the training stage. In this case, we need an algorithm which can handle data in the order of their submission. A sequential learning variant of ELM, known as OS-ELM, is suitable in such situations. The use of OS-ELM is an attempt to verify whether the sequential learning can improve the recognition rate. OS-ELM provides memory economy to the pattern recognition system. It produced better result than E-ELM. The classifier gave a recognition rate of 67.33% (db2 at level 3) to 90.69% (db6 at level 6) for different wavelets. It spend more time (0.34 s (db2 at level 3) to 0.75 s (db2 at level 6)) for testing than E-ELM, whereas the training time varies from 45 s (db2 at level 3) to 204 s (db6 at level 6). The number of hidden neurons in the network is determined through experimentation. The best performing measures for each feature set is shown in Table 1. Since the training observations of a particular block are discarded immediately after its use, OS-ELM supports more hidden nodes and training patterns. In this experiment, an initial block of 4,000 patterns is used, and further blocks ranged from 200 to 400 patterns. The highest recognition accuracy of 90.69% is obtained for db6 wavelet at sixth decomposition level by spending 0.66 s in testing. In E-ELM and OS-ELM, the training time increased with the increase of hidden neurons.

The classification results using ELM show a clear distinction with its variants. Based on the experiments, the

number of hidden neurons is determined as 2500 for all feature sets. Since the hidden layer size is same, the variation in different training times [204 s (db4 at level 6) to 216 s (db2 at level 6)] and testing times [0.83 s (sym3 at level 6) to 1.06 s (db2 at level 3)] is also less. The testing time is more than OS-ELM due to the large size of hidden layer. But, at the same time, ELM is more time efficient than its variants for the same number of hidden neurons. The recognition accuracy obtained [90.02% (db2 at level 3) to 95.59% (db6 at level6)] is also more than that of E-ELM and OS-ELM. Even though OS-ELM can accommodate more hidden neurons, it does not improve the recognition accuracy beyond a certain limit. This is due to less number of training patterns handled at any stage of its learning. For this character recognition problem, db6 wavelet is found to be promising. WEF using db6 at the sixth decomposition level achieved the highest recognition rate of 95.59% by spending 212 s for training and 0.89 s for testing. ELM showed consistent performance with all wavelets and at every level of decomposition. The generalization performance of this algorithm is very stable on a wide range of hidden nodes, although it tends to become worse when too few or too many nodes are randomly generated. ELM appears to be suitable in applications which require fast prediction and response capability [16].

Table 1 Recognition accuracy (%) using different wavelets

Wavelet	Level	ELM	E-ELM	OS-ELM	
				Accuracy	Hidden neurons
db2	3	90.02	76.54	67.33	1,000
db2	4	91.15	73.37	80.29	1,500
db2	5	93.19	74.90	83.90	1,500
db2	6	95.33	73.54	87.18	1,750
db3	6	94.21	74.98	85.90	1,500
db4	6	90.79	68.14	76.96	1,200
db5	6	95.17	76.84	88.90	1,500
db6	6	95.59	75.71	90.69	1,750
bior1.3	6	90.67	66.52	76.34	1,200
bior1.5	6	91.82	68.26	78.09	1,200
sym3	6	94.16	75.29	85.79	1,500

The recent works on Malayalam character recognition with the same database has achieved much improvement in recognition accuracy. Bindu and Raju [87] compared the performance of QDF and MQDF using NVD, run length coding (RLC) [88], and gradient features [89]. MQDF showed the highest recognition rate of 92% (NVD), 94.18% (RLC from thinned image), and 94.96% (gradient feature) in the respective feature sets. When the RLC is extracted from edge image, the recognition accuracy is increased to 95.16% using MLP [90]. From these results, it can be seen that ELM is able to produce the highest recognition rate in this classification problem. MQDF is quick in training, whereas it took 168 s to test the gradient feature set of size 35. Both MLP and ELM used a feature set of size 18. MLP is fast in testing, but it too slow in training. ELM does not have the problem of slow testing or training.

6 Conclusion

The handwriting recognition has been a subject of intensive research for the past few decades. The slow speed of leaning algorithms was a major hurdle for these experiments. In this paper, we have overcome this difficulty using ELM. The leaning speed of ELM is extremely fast, and it produced only the smallest training error. It can be used to train SLFN with non differentiable activation functions. The feature vector used for this pattern classification problem is based on wavelet energy. WEF is defined by employing wavelets, which is a powerful tool for multi-resolution analysis. It can reflect the wavelet energy distribution of characters in several directions at different wavelet decomposition levels. The discriminability of each level WEF is also investigated and it shows that the order of these discriminabilites increase as the level increase. Usually, most of the people keep their paper in an angular position while writing. As a result a skewness is created in most of the characters. Since WEF is robust to some extent of rotation and translation of the images, we have not used any skew correction techniques in this work. Since the three wavelets have shown good performance in this pattern recognition problem, there is a scope for further improvement. The future research is in the direction of employing statistical measures on wavelet coefficients to extract more discrimating features from character images. Based on the recent works on Malayalam character recognition, it is required to conduct experiments using different features with better performing classifiers.

Acknowledgments The authors are grateful to reviewers who have helped us to prepare this paper in the present form. Also, the authors thank University Grants Commission and Kerala State Council for Science, Technology and Environment for providing fellowship to

Binu P. Chacko and Vimal Krishnan V R respectively, to carry out this research work.

References

1. Santhosh KC, Nattee C (2009) A comprehensive survey on on-line handwriting recognition technology and its application to the Nepalese Natural Handwriting. *Kathmandu Univ J Sci Eng Technol* 5(1):31–55
2. Plotz T, Fink GA (2009) Markov models for offline handwriting recognition: a survey. *Int J Document Anal Recogn* 12:269–298
3. Raju G (2008) Wavelet transform and projection profiles in handwritten character recognition—a performance analysis. In: *Proceedings of International Conference on ADCOM*, pp 309–313
4. Plamondon R, Srihari SN (2000) On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans Pattern Anal Mach Intell* 22(1):63–84
5. Bhattacharya U, Chaudhuri BB (2009) Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. *IEEE Trans Pattern Anal Mach Intell* 31(3):444–457
6. Liu C-L, Nakashima K, Sako H, Fujisawa H (2003) Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recogn* 36:2271–2285
7. Kimura F, Miyake Y, Sridhar M (1995) Handwritten ZIP code recognition using Lexicon free word recognition algorithm. *Proc Int Conf Document Anal Recogn* 2:906–910
8. Liu H, Ding X (2005) Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes. In: *Proceedings of International Conference on Document Analysis and Recognition*, pp 19–25
9. Shi M, Fujisawa Y, Wakabayashi T, Kimura F (2002) Handwritten numeral recognition using gradient and curvature of gray scale image. *Pattern Recogn* 35(10):2051–2059
10. Shustorovich A (1994) A subspace projection approach to feature extraction: the 2D gabor transform for character recognition. *Neural Netw* 7(8):1295–1301
11. Heutte L, Paquet T, Moreau JV, Lecourtier Y, Oliver C (1998) A structural/statistical feature based vector for handwritten character recognition. *Pattern Recogn Lett* 19(7):629–641
12. Wu X-Q, Wang K-Q, Zhang D (2005) Wavelet energy feature extraction and matching for palmprint recognition. *J Comput Sci Technol* 20(3):411–418
13. Juang C-F, Cheng C-N, Chen T-M (2009) Speech detection in noisy environments by wavelet energy-based recurrent neural fuzzy network. *Expert Syst Appl* 36:321–332
14. Huang G-B, Chen L, Siew C-K (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans Neural Netw* 17(4):879–892
15. Zhu Q-Y, Qin AK, Suganthan PN, Huang G-B (2005) Evolutionary extreme learning machine. *Pattern Recogn* 38:1759–1763
16. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70:489–501
17. Lorigo LM, Govindaraju V (2006) Offline arabic handwriting recognition: a survey. *IEEE Trans Pattern Anal Mach Intell* 28(5):712–724
18. Srihari SN, Yang X, Ball GR (2007) Offline Chinese handwriting recognition: an assessment of current technology. *Front Comput Sci China* 1(2):137–155
19. Abuhaiba ISI, Mahmoud SA, Green RJ (1994) Recognition of handwritten cursive Arabic characters. *IEEE Trans Pattern Anal Mach Intell* 16:664–672
20. Amin A, Al-Sadoun H, Fischer S (1996) Hand-printer arabic character recognition system using an artificial network. *Pattern Recogn* 29:663–675

21. Makhoul J, Schwartz R, Lapre C, Bazzi I (1998) A script independent methodology for optical character recognition. *Pattern Recogn* 31:1285–1294
22. Khorshed MS, Clocksin WF (1999) Structural features of cursive arabic script. In: *Proceedings of British Machine Vision Conference*, pp 422–431
23. Al-Shaher AA, Hancock ER (2003) Learning mixtures of point distribution models with the EM algorithm. *Pattern Recogn* 36:2805–2818
24. Deghani A, Shabani F, Nava P (2001) Offline recognition of isolated Persian handwritten characters using multiple hidden markov models. In: *Proceedings of international conference on information technology: coding and computing*, vol 20, pp 506–510
25. Gillevic D, Suen CY (1998) HMM-KNN word recognition engine for bank cheque processing. In: *Proceedings of International Conference on Pattern Recognition, Brisbane, Australia, August 1998*, pp 1526–1529
26. Kennr S, Anisimov V, Baret O, Gorski N, Price D, Simon IC (1997) The A2iA intercheque system: courtesy amount and legal amount recognition for French cheques. In: *Impedovo S, Wang PSP, Bunke H (eds) Machine perception and artificial intelligence*, vol 28. World Scientific, USA, pp 43–86
27. Kim G, Govindaraju V (1997) Bank cheque recognition using cross validation between legal and courtesy amounts. *Int Conf Pattern Recogn Artif Intell* 11(4):657–674
28. Ko AH, Cavalin PR, Sabourin R, de Souza Britto A Jr (2009) Leave-one-out-training and leave-one-out-testing hidden markov models for a handwritten numeral recognizer: the implications of a single classifier and multiple classifications. *IEEE Trans Pattern Anal Mach Intell* 31(12):2168–2178
29. Nunes CM, de S Britto Jr. A, Kaestner CAA, Sabourin R (2004) An optimized hill climbing algorithm for feature subset selection evaluation on handwritten character recognition. In: *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, pp 365–370, 2004
30. Giusti N, Masuli F, Sperduti A (2002) Theoretical and experimental analysis of a two stage system for classification. *IEEE Trans Pattern Anal Mach Intell* 24(7):893–904
31. Cao J, Ahmadi M, Sridhar M (1995) Recognition of handwritten numerals with multiple features and multistage classifiers. *Pattern Recogn* 28(2):153–160
32. Vamvakas G, Gatos B, Perantonis SJ (2010) Handwritten character recognition through two-stage foreground sub-sampling. *Pattern Recogn* 43:2807–2816
33. Huang L, Liu C (2010) Handwritten Chinese character recognition method based on non-parametric dimensionality reduction. In: *Proceedings of International Conference on Computer Design and Applications*, pp VI-217–VI-220
34. Tsukumo J, Tanaka H (1988) Classification of hand-printed Chinese characters using non linear normalization and correction methods. In: *Proceedings of international conference on pattern recognition, Roma, Italy*, pp 168–171
35. Fu H-C, Xu Y-Y (1998) Multilingual handwritten character recognition by Bayesian decision based neural networks. *IEEE Trans Signal Process* 46(10):2781–2789
36. Mao J, Mohiuddin KM (1997) Improving OCR performance using character degradation models and boosting algorithm. *Pattern Recogn Lett* 18:1415–1419
37. Dong J-X, Krzyzack A, Suen C-Y (2005) Fast SVM training algorithm with decomposition on very large training sets. *IEEE Trans Pattern Anal Mach Intell* 27(4):603–618
38. Liu C-L, Sako H, Fujisawa H (2003) Handwritten Chinese character recognition: alternatives to non linear normalization. In: *Proceedings of international conference on document analysis and recognition, Edinburgh, Scotland*, pp 524–528
39. Liu C-L, Marukawa K (2005) Pseudo 2D shape normalization methods for handwritten Chinese character recognition. *Pattern Recogn* 38(12):2242–2255
40. Liu C-L, Sako H, Fujisawa H (2002) Performance evaluation of pattern classifiers for handwritten character recognition. *Int J Document Anal Recogn* 4:191–204
41. Kato N, Suzuki M, Omachi S, Aso H, Nemoto Y (1999) A handwritten character recognition system using directional element feature and symmetric Mahalanobis distance. *IEEE Trans Pattern Anal Mach Intell* 21(3):2781–2789
42. Leung K-C, Leung CH (2009) Recognition of Chinese handwritten characters by combining regularization, Fisher's discriminant and distorted sample generation. In: *Proceedings of International Conference on Document Analysis and Recognition*, pp 1026–1030, Barcelona
43. Leung KC, Leung CH (2010) Recognition of handwritten Chinese characters by critical region analysis. *Pattern Recogn* 43:949–961
44. Chinnuswamy P, Krishnamoorthy SG (1980) Recognition of hand-printer Tamil characters. *Pattern Recogn* 12(3):41–152
45. Shanthi N, Duraiswamy K (2009) A Novel SVM-based handwritten Tamil character recognition. *Pattern Anal Appl*
46. Sukhaswami MB, Seetharamulu P, Pujari AK (1995) Recognition of Telugu characters using neural networks. *Int J Neural Syst* 6(3):317–357
47. Desai AA (2010) Gujarati handwritten numeral optical character recognition through neural network. *Pattern Recogn* 43:2582–2589
48. Chaudhuri BB, Pal U (1997) Skew angle detection of digitized Indian script documents. *IEEE Trans Pattern Anal Mach Intell* 19(2):182–186
49. Rajasekararadhya SV, Ranjan PV (2009) Efficient zone based feature extraction algorithm for handwritten numeral recognition of popular south Indian scripts. *J Tech Appl Inform Technol* 7(1):1171–1180
50. Dutta A, Chaudhary S (1993) Bengali Alpha numeric character recognition using curvature features. *Pattern Recogn* 26(12):1757–1770
51. Bhattacharya U, Das TK, Datta A, Parui SK, Chaudhuri BB (2002) A hybrid scheme for hand-printed numeral recognition based on a self organizing network and MLP classifiers. *Int J Pattern Recogn Artif Intell* 16(7):845–864
52. Bhowmick TK, Bhattacharya U, Parui SK (2004) Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. In: *Proceedings of international conference on neural information processing*, pp 814–819
53. Hanmandlu M, Murthy OVR (2007) Fuzzy model based recognition of handwritten numerals. *Pattern Recogn* 40(6):1840–1854
54. Banashree NP, Andhre D, Vasanta R, Satyanarayana PS (2007) OCR for script identification of Hindi (Devnagari) numerals using error diffusion Halftoning algorithm with neural classifier. *Proc World Acad Sci Eng Technol* 20:46–50
55. Lajish VL (2007) Adaptive neuro-fuzzy inference based pattern recognition studies on handwritten character images. Ph.D dissertation, University of Calicut, India
56. Raju G, Moni BS (2009) Global and local elastic meshing for handwritten Malayalam character recognition. *Int J Comput Inform Technol Eng* 3(1):149–153
57. Moni BS, Raju G (2009) Meshing and normalized vector distance from centroid for handwritten Malayalam character recognition. In: *Proceedings of ICSIP*, pp 398–403
58. Moni BS, Raju G (2009) Multiple MLP classifiers for handwritten Malayalam character recognition. In: *Proceedings of IC-MCM*, pp 349–354
59. Moni BS, Raju G (2010) Quadratic classifier for handwritten Malayalam character recognition. In: *Proceedings of NCSC*, pp 59–68

60. Raju G, Moni BS (2009) Global elastic meshing for handwritten Malayalam character recognition. In: Proceedings of national conference on computer science and engineering, pp 10–14
61. Chacko BP, Babu Anto P (2009) Discrete curve evolution based skeleton pruning for character recognition. In: Proceedings of IAPR, pp 402–405
62. Chacko BP, Vimal Krishnan VR, Babu Anto P (2010) Character recognition using multiple back propagation algorithm. In: Proceedings of national conference on image processing, pp 209–212
63. John R, Raju G, Guru DS (2007) 1D wavelet transform of projection profiles for isolated handwritten character recognition. In: Proceedings of ICCIMA, vol. 2, pp 481–485
64. Wei XK, Li YH, Feng Y (2006) Comparative study of extreme learning machine and support vector machine, advances in neural networks, pp 1089–1095, 2006
65. Mahmoud SA, Olatunji SO (2009) Automatic recognition of offline handwritten Arabic (Indian) numerals using support vector machine and extreme learning machine. *Int J Imaging* 2(A09): 34–53
66. Li M, Wang C, Dai R (2008) Unconstrained handwritten character recognition based on WEDF and multilayer neural network. In: Proceedings of World Congress on Intelligent Control and Automation, pp 1143–1148
67. Li H-N, He X-Y, Yi T-H (2009) Multi-component seismic response analysis of offshore platform by wavelet energy principle. *Cost Eng* 56:810–830
68. Achuthan A, Rajeswari M, Ramachandram D, Aziz ME, Shuaib IL (2011) Wavelet energy-guided level set-based active contour: a segmentation method to segment highly similar regions. *Comput Biol Med* (in press)
69. Favata J, Srikanth G, Srihari S (1994) Hand-printer character/digit recognition using a multiple feature/resolution philosophy. In: Proceedings of international workshop on frontiers of handwriting recognition, pp 57–66
70. Park J, Govindaraju V, Srihari SN (2000) OCR in a hierarchical feature space. *IEEE Trans Pattern Anal Mach Intell* 2(4):400–407
71. Raju G (2006) Recognition of unconstrained handwritten Malayalam characters using zero crossing of wavelet coefficients. In: Proceedings of international conference on Advanced Computing and Communications, pp 217–221
72. Sasi S, Schwiebert L, Bedi JS (2007) Wavelet packet transform and neuro-fuzzy approach to handwritten character recognition
73. Lee S-W, Kim C-H, Tang YY (1996) Multi-resolution recognition of unconstrained handwritten numerals with wavelet transform and multilayer cluster neural network. *Pattern Recogn* 29(12):1953–1961
74. Correia SEN, Carvalho JM (2000) Optimizing the recognition rates of unconstrained handwritten numerals using biorthogonal spline wavelets, vol 2. ICPR, Barcelona, Spain, pp 2251
75. Huang G-B, Babri HA (1998) Upper bounds on the number of hidden neurons in feed forward with arbitrary bounded non linear activations functions. *IEEE Trans Neural Netw* 9(1):224–229
76. Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feed forward networks. *IEEE Trans Neural Netw* 17(6):1411–1423
77. Suresh S, Saraswathi S, Sundararajan N (2010) Performance enhancement of extreme learning machine for multi-category sparse data classification problems. *Eng Appl Artif Intell* 23:1149–1157
78. Suresh S, Venkatesh Babu R, Kim HJ (2009) No-reference image quality assessment using modified extreme learning machine classifier. *Appl Soft Comput* 9:541–552
79. Rong HJ, Ong YS, Tan AW, Zhu Z (2008) A fast pruned extreme learning machine for classification problem. *Neurocomputing* 72:359–366
80. Miche Y, Sorjamma A, Bas P, Simula O, Jutten C, Lendasse A (2010) OP-ELM: optimally pruned extreme learning machine. *IEEE Trans Neural Netw* 21(1):158–162
81. Lan Y, Soh YC, Huang GB (2010) Constructive hidden nodes selection of extreme learning machine for regression. *Neurocomputing* 73(16–18):3191–3199
82. Lan Y, Soh YC, Huang GB (2010) Two stage extreme learning machine for regression. *Neurocomputing* 73:3028–3038
83. Huang G-B, Chen L (2007) Convex incremental extreme learning machine. *Neurocomputing* 70:3056–3062
84. Barron AR (1993) Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans Inf Theory* 39(3):930–945
85. Huang G-B, Chen L (2008) Enhanced random search based incremental extreme learning machine. *Neurocomputing* 71: 3460–3468
86. Feng G, Huang G-B, Lin Q, Gay R (2009) Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Trans Neural Netw* 20(8):1352–1357
87. Moni BS, Raju G (2011) Modified quadratic classifier and normalized vector distance for handwritten Malayalam character recognition. In: Proceedings of international conference on emerging trends in mathematics and computer applications, pp 356–360, December 2011
88. Moni BS, Raju G (2011) Modified quadratic classifier for handwritten Malayalam character recognition using run length count. In: Proceedings of international conference on emerging trends in electrical and computer technology
89. Moni BS, Raju G (2011) Modified quadratic classifier for handwritten character recognition using gradient features. In: Proceedings of Natural Conference on Indian Language Computing
90. Chacko BP, Babu Anto P (2010) Pre and post processing approaches in edge detection for character recognition. In: Proceedings of international conference on frontiers in handwriting recognition, pp 676–681