



Improving the Accuracy for Offline Arabic Digit Recognition Using Sliding Window Approach

Ebrahim Al-wajih^{1,2} · Rozaida Ghazali¹

Received: 13 September 2018 / Accepted: 18 January 2020 / Published online: 28 January 2020
© Shiraz University 2020

Abstract

Handwritten Digit recognition is a challenging problem these days due to the widely used Arabic language in the world, especially in the Middle East region. In this paper, sliding windows are used to enhance classification accuracies and implemented using random forests (RF) and support vector machine (SVM) classifiers for recognition of Arabic digit images. In order to study their effectiveness with and without using sliding windows, four different feature extraction techniques have been proposed which includes Mean-based, Gray-Level Co-occurrence Matrix (GLCM), Moment-based, and Edge Direction Histogram (EDH). The obtained accuracies show the significance of using sliding windows for classifying digit. The recognition rates acquired using the modified version of AHDBase dataset are 98% when Mean-based and Moment-based are applied with RF classifier, 98.33% and 99.13% when GLCM and EDH are used with linear-kernel SVM, respectively. Moreover, the performance of this study is compared against recent state-of-the-art approaches, namely Geometric-based, two-dimensional discrete cosine transform, Hierarchical features, Hetero-features, Discrete Fourier Transform and geometrical features, Gabor-based, gradient, structural, and concavity and Local Binary Convolutional Neural Networks.

Keywords Arabic digit recognition · Sliding window · Texture features · Gray-level co-occurrence matrix · Edge direction histogram

1 Introduction

Handwritten digit recognition is still a challenging problem although many studies have been carried out to address it. This is due to the fact that the exponential development of the technology, especially with the smart devices, needs noncomplex algorithms to enhance the accuracy of recognition systems. Further, it is difficult to consider that the problem of identifying numbers was solved because of the limited sizes of datasets which is the common issue in

pattern recognition area. The number of writers in the available datasets does not cover all variations of writing styles from person to person. To solve this problem, several different techniques have been proposed by researchers to find the appropriate algorithm. Handwritten digit recognition is categorized into two kinds, online and offline, based on the input method to the system. The applications that used online method receive the input by the movement of the pen on a pen-based screen, while the offline applications use the image of a digit that is captured using an interface such as a scanner or camera (Plamondon and Srihari 2000). Many applications need an automatic recognition system to recognize Digit with high accuracy and speed such as (e.g., postal code, bank checks reading in the offline system and editors, enjoyment applications in online systems).

The form of the digit varies from one language to another, which in turn led to the need of constructing different handwriting recognition systems. English Digits recognition has been examined in many studies before four

✉ Ebrahim Al-wajih
ebrahim.q.alwajih@gmail.com

Rozaida Ghazali
rozaida@uthm.edu.my

¹ Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Batu Pahat,
86400 Parit Raja, Johor, Malaysia

² Society Development & Continuing Education Center,
Hodeidah University, Alduraimi, 3114, Hodeidah, Yemen



decades (Trier et al. 1996), while Arabic digit has been investigated in the nineties. After that, many studies on Arabic handwriting recognition have been carried out using different Arabic handwritten datasets. Arabic and English handwriting recognition is a challenging problem because the writing style differs from writer to others as well as the variation of style at different instances of the same writer. Moreover, in Arabic or other languages, some digit's form is changed to be almost similar to another digit's form due to the rotation issues such as seven and eight digits in Arabic language or six and nine in the Latin language. In addition, some of the digits have convergent forms such as two and six in Arabic when the digit image is flipped. Due to this, handwriting recognition is considered as one of the challenging problems in machine learning applications. In previous works (Awaida and Mahmoud 2014; El-Sherif and Abdelazeem 2007; Ghaleb et al. 2013; Lawgali 2015; Mahmoud and Al-Khatib 2011; Rashnodi et al. 2011), the classification accuracy has been improved due to the significance of proposed features. However, selecting feature extraction techniques is the big challenge to acquire high recognition rate including those that need such complex implementation such as Gabor-filters-based (Haghighat et al. 2013), GIST (Oliva and Torralba 2001), and pyramid histogram of oriented gradients (PHOG) (Bosch et al. 2007) features.

A digital image is a grid of pixels, and each pixel includes a value called pixel intensity. Basically, images are represented using a 2D plane in terms of a space called spatial domain. On the other hand, when the pixel values are modified using any methods, such as the Fourier transform, then the domain of the modified image is changed and called frequency domain. In this paper, a complexity term is used to determine whether the features are extracted from the spatial domain or frequency domain. When the features are extracted from spatial domain, the cost of implementation or the complexity decreases, while if they are extracted from the frequency domain the cost of implementation or the complexity increases due to the transformation step. Moreover, if the features are not extracted directly from the pixel intensity of images, they are considered like those extracted from the frequency domain. This complexity motivates us to propose a model with less complexity as well as an acceptable high accuracy. To this end, the proposed features are extracted using a sliding window to obtain more informative features than those extracted using the whole image (global level). These windows can reduce the similarities occurred due to the rotation issues of writers.

In this paper, sliding window approach has been proposed to increase the accuracy as well as to reduce such complexity that occurs due to the transformation step. The experiments of this study show that the obtained accuracies

have been significantly increased using this approach, while the accuracies obtained without sliding window are very poor. Feature extraction techniques of simple and complex implementation such as Mean-based, Gray-Level Co-occurrence Matrix (GLCM), Moment-based, and Edge Direction Histogram (EDH) have been explored to show the significance of using the sliding windows as well as the significance of features and classifiers. In addition, the proposed approach has been validated using four datasets and compared against state-of-the-art techniques such as Local Binary Convolutional Neural Networks (LBCNN).

The rest of the paper is organized as follows. In Sects. 2 and 3, related works and the proposed methods are described, respectively; after that setup and the design of the experiments are illustrated in Sect. 4. The results are discussed and compared against the state-of-the-art digit recognition systems in Sects. 5 and 6, respectively. At the end, conclusion and future work are presented in Sect. 7.

2 Related Work

Many studies have been carried out in Arabic Handwritten Digit recognition. In this section, some of the recent studies related to Arabic Handwritten Digit recognition are discussed.

A feature extraction technique based on two-dimensional discrete cosine transform (2D DCT) coefficients was proposed (AlKhateeb and Alseid 2014). The DCT coefficients were sorted in the zigzag order, and the first 20 coefficients were chosen to build the model. This approach was investigated using a subset of AHDBase database (500 digits). Lawgali suggested a feature extraction technique based on discrete wavelet transform (DWT) and DCT. The Haar wavelet was selected to generate the low-frequency coefficients of DWT image, and then the coefficients of DCT were extracted from the low-frequency coefficients. The DCT coefficients were also sorted in the zigzag order, and the first 50 coefficients were chosen to train the model (Lawgali 2015). Meanwhile, the Discrete Fourier Transform (DFT) and geometrical features were used to extract features (Rashnodi et al. 2011). Other features were suggested using Log-Gabor filters (Mahmoud and Al-Khatib 2011). The mean and variance were computed from the blocks, created using the sliding window technique, to each produced energy of Log-Gabor filter. Different classifiers were used to explore this approach, and the highest accuracy was obtained using SVM. However, all of these studies extracted features from the frequency domain.

A rough neural network was applied to build a hybrid model to address the proneness to the overfitting problem and the empirical nature of the model development problem in neural networks (Radwan 2013). In Radwan work,



Digit was represented using the typewritten digits by 11 features and then reduced to 5 features using discernibility matrix. The reduced features were clustered and applied to rough neural network (RNN) that is considered as a kind of complex implementation, or it is considered as a frequency domain.

An approach based on hybrid classification was developed (Takruri et al. 2015). The pixel intensity was used as a feature with two hybrid classifiers (serial hybrid classifier and parallel hybrid classifier). The serial hybrid classifier includes three levels of single classifier, Fuzzy C-Means classifier, Support Vector Machine and a unique pixel method which forms the third classification level. In parallel hybrid classifier, the final decision was produced using the Fuzzy C-Means classifier with a Neural Network simultaneously. Other features were extracted from Digit based on five properties of an image: the area, center, perimeter, diameter, and the bounding box (Shilbayeh et al. 2013). The Digit image was divided into 3×3 blocks. After that, the features were extracted from each region with different sizes and then normalized to be 18 features. Although both of these proposed features extracted from the spatial domain, their accuracies are not good enough.

The Digit features using the percentage of strokes in both horizontal and vertical directions and some morphological operations were suggested using some rules (Ghaleb et al. 2013). The rules were created in the decision-making step. This step contains three sub-steps, Grand Class Type Decision, Single Numeric Class Decision, and Unrecognized Numerals Decision. In addition, used hetero-features were applied to the private dataset to improve the recognition rate (El-Sherif and Abdelazeem 2007). The proposed feature is based on a contour following algorithm, the number of pixels, the number of transitions, and centroidal distances. However, both of them used hydrogenase of features to get the high accuracies. Another feature extraction technique based on gradients, structural, and concavity (GSC) features was proposed to capture narrow, intermediate, and large-scale qualities of the digits (Awaida and Mahmoud 2014).

All previous works described in this section did not apply sliding windows approach except Mahmoud and Al-Khatib. However, they used in a frequency domain that increases the complexity of extracting features.

3 Proposed Models

In this paper, a digit recognition model is proposed by applying several sizes of sliding windows in order to improve the performance significantly. Mean-based, GLCM-based, Moment-based, and edge direction histogram have been applied as feature extraction techniques, and two classifiers are utilized. Exploring the significance

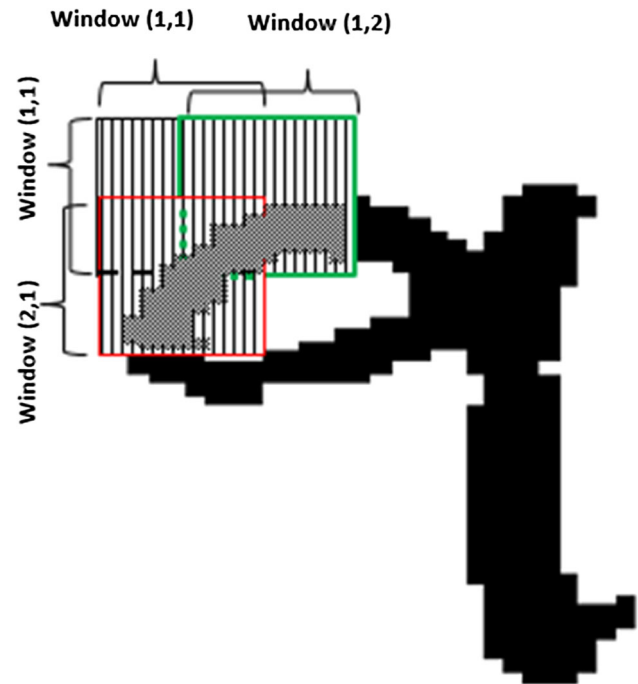


Fig. 1 An example use of sliding window for digit

of all of them can help the researchers or manufacturers develop an optimal mixture between feature extraction techniques, sliding window size, and classification approaches, that have been methodically investigated, for desirable digit recognition systems.

3.1 Sliding Windows

One of the main proposed methods in this study is the sliding windows/frames. The sliding window is a virtual window or a rectangular region of a fixed $Q \times L$ size that slides vertically or horizontally across an image. Four different sizes of sliding windows of 16×16 , 16×8 , 8×16 , and 8×8 pixels have been utilized. A sliding window of 50% overlap has been passed vertically and horizontally generating local regions that have been used to extract features. The number of the produced local regions NLR is calculated as the following:

$$NLR = \left[\frac{M}{Q * 0.5} - 1 \right] * \left[\frac{N}{L * 0.5} - 1 \right] \quad (1)$$

where M and N are the width and the height of an image, Q and L are the width and the height of a sliding window. Figure 1 shows an example of applying sliding windows.

3.2 Feature Extraction

In real life, each object has its attributes called features that can distinguish it from other objects. In pattern recognition

systems, these features are extracted from the images of the object using several techniques as used in previous works (Awaida and Mahmoud 2014; Bosch et al. 2007; El-Sherif and Abdelazeem 2007; Ghaleb et al. 2013; Haghighat et al. 2013; Lawgali 2015; Mahmoud and Al-Khatib 2011; Oliva and Torralba 2001; Rashnodi et al. 2011). Four feature extraction techniques are proposed and detailed in the following subsections.

3.2.1 Mean-Based Features

For Mean-based features approach, the arithmetic mean is computed from each block created using sliding windows as:

$$\text{mean} = \frac{\sum_{i=1}^n \sum_{j=1}^m x_{i,j}}{n * m} \quad (2)$$

where n and m are the height and width of a created block, respectively. i and j are the coordinates (row and column) of a pixel in the block, and $x_{i,j}$ is the value of the corresponding pixels. The number of the extracted features is 9, 21, 21, and 49 for 16×16 , 16×8 , 8×16 and 8×8 windows, respectively. It is clear that the implementation of this technique is simple.

3.2.2 GLCM-Based Features

In this approach, the texture features are extracted based on energy, entropy, contrast, difference of variance, dissimilarity, sum of entropy and sum of average. These kinds of features can capture the characteristics of the image parts with respect to changes in certain directions and the scale of the changes. Gray-level co-occurrence matrix (GLCM) (Haralick et al. 1973) is used widely to analyze texture features. GLCM can estimate the image characteristics with respect to changes in certain directions and the scale of the changes. “Each entry (i, j) in GLCM represents the number of occurrences of the pair of gray levels i and j which are a distance d apart in original image” (Partio et al. 2002). Haralick proposed several statistical features extracted from GLCM to assess the relation between different gray-level co-occurrence matrices (Haralick et al. 1973). From these features, only seven are selected in this study formulated as:

(a) Energy:

$$\sum_i \sum_j p(i, j)^2 \quad (3)$$

(b) Entropy:

$$\sum_i \sum_j p(i, j) \cdot \log(p(i, j)) \quad (4)$$

(c) Contrast:

$$\sum_i \sum_j (i - j)^2 \cdot p(i, j) \quad (5)$$

(d) Dissimilarity:

$$\sum_i \sum_j |i - j| \cdot p(i, j) \quad (6)$$

(e) Difference of variance:

$$\sum_{i=0}^{N_g-1} i^2 p_{x-y}(i) \quad (7)$$

(f) Sum of entropy:

$$- \sum_{i=2}^{2N_g} p_{x+y}(i) \cdot \log\{p_{x+y}(i)\} \quad (8)$$

(g) Sum of average:

$$\sum_{i=2}^{2N_g} i \cdot p_{x+y}(i) \quad (9)$$

where N_g is the number of gray levels found in the input, $P(i, j)$ is the (i, j) th entry in the GLCM, x and y are the coordinates (row and column) of an entry in the co-occurrence matrix, $p_{x-y}(i)$ and $p_{x+y}(i)$ are the probability of co-occurrence matrix coordinates summing to $x - y$ and $x + y$, respectively. Due to that the features are extracted after creating the gray-level co-occurrence matrix from image, it can be considered as an implemented complex feature.

3.2.3 Moment-Based Features

Hu moment invariant method is used to generate the shape descriptor in this study. According to (Hu 1962), relative and absolute combinations of moments were produced that are invariant even if the scale, position, and orientation are changed. The features are extracted from the intensity of images as the following, for each image of $M \times N$ pixels, Hu moment function m_{pq} with $(p + q)$ order moment is calculated as:

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p \cdot y^q \cdot f(x, y) \quad (10)$$

where $p, q = 0, 1, 2, \dots$ are integers.

To find the coordinates of the centers of mass for each image, the following equation is used:

$$x' = \frac{M_{10}}{M_{00}}, \quad y' = \frac{M_{01}}{M_{00}} \quad (11)$$

If translation, rotation, or scale makes an effect of an image, then the image may be positioned such that its

center of mass coincided with the origin of the field of view, that is $(x = 0)$ and $(y = 0)$. The computed moments by the center of mass are referred to as a central moment (Gonzalez and Woods 2008) formulated as:

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - x')^p \cdot (y - y')^q \cdot f(x, y) \quad (12)$$

The central moment is normalized by a scaling normalization as:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}}. \quad (13)$$

where $\gamma = (p + q)/2 + 1$ is the normalization factor.

Eight invariant moments are calculated based on the normalized central moments of order three. These moments are invariant even if the scale, translation, and rotation of image are changed. The invariant moments $\phi_1, \phi_2, \dots, \phi_7$ were proposed by Hu (1962), and the last moment ϕ_8 used in this study was proposed by Flusser and Suk (2006) as:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11} \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \cdot [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03}) \cdot (\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ &\quad + 4(\eta_{30} + \eta_{12}) \cdot (\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \cdot [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (\eta_{30} - 3\eta_{12}) \cdot (\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] \\ \phi_8 &= \eta_{11}(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2 \\ &\quad - (\eta_{20} - \eta_{02}) \cdot (\eta_{30} + \eta_{12}) \cdot (\eta_{03} + \eta_{21}) \end{aligned}$$

3.2.4 Edge Direction Histogram

Gao et al. (2008) proposed edge direction histogram (EDH) as features to represent the shape of images. In this technique, vertical and horizontal edge maps are calculated using masks. $[-1, 0, 1]$ and $[-1, 0, 1]^T$ have been used in this study as masks. Assume v and h to be the vertical and horizontal edge values that are computed by convoluting the edge masks with the original image, respectively. The edge map is computed by $\tan^{-1} \frac{v}{h}$, and the magnitude, where the features are extracted from, is calculated using $m = \sqrt{v^2 + h^2}$. An interval of 18° is applied to discretize the obtained edge map. Then the histogram of the pixel magnitude values is computed to obtain the feature vector.

3.3 Classifier Techniques

Many techniques were used to classify pattern classes. In this study, two common classifiers are applied, including random forest classifier as a tree-based ensemble classifier, and support vector machine as a kernel-based classifier.

3.3.1 Random Forests Classifier

Random Forest classifier is a large collection of tree classifiers (Breiman 2001). The idea behind this classifier is by averaging noisy and unbiased models to build models with low variance, in terms of classification. Each tree classifier is grown in random form.

Let $D = \{(x_1, y_1), (x_n, y_n)\}$. are the samples used to build a model T , $x_i \in \mathbb{R}^d$, D_i are bootstraps samples from D , $i = 1, \dots, B$, T_i is a tree built using D_i such that at each node in T_i a random subset of m features is chosen, $m \leq d$, and then split into two daughter nodes, the first node has the best feature and the second node is recursively split. At the end of the building model, B trees are constructed, and all trees are trained independently.

Let v be a test point needed to be classified. This point is passed through all trees starting from root until it reaches the corresponding leaf. Due to that, each tree is created using a different dataset, and the probability of the posteriors of each tree has an important rule to make a decision. These probabilities are computed as follows: Let C be a set of classes and L a set of leaves for T_i . In the training process, the probabilities of posteriors $P_{T_i,l}(Y(v) = c)$ for each class $c \in C$ at each leaf node $l \in L$ are computed for each tree $T_i \in T$. These probabilities are computed by finding the ratio of the number of samples of class c that reach l to the total number of samples that reach l . $Y(v)$ is the class-label c for sample v .

3.3.2 Support Vector Machine Classifier (SVM)

Support vector machine is a classification approach that is used to classify linear or nonlinear data. The first work using SVM was proposed by Boser et al. (1992). The idea of using SVM is based on statistical learning theory (Vapnik and Chervonenkis 1971). In general, the idea of this classifier is by separating dataset of two classes with a maximum distance between them. The performance of the SVM classifier differs based on a kernel function and the data of the problem itself. The data that are used to build a model can be linearly separable or nonlinear separable. The best kernel function is used with SVM based on the nature of data, linear separable or nonlinear separable. There are many kernel functions used with SVM classifier such as linear, polynomial, sigmoid and radial basis function

(RBF) kernels. In this study, RBF and linear kernel functions are used to find the best SVM-based systems, in which if the data are linear separable, then the linear kernel would perform better than RBF. Meanwhile, if the data are nonlinear separable, RBF kernel would improve the performance of the proposed systems. For training SVM, Sequential Minimal Optimization (SMO) (Keerthi et al. 2001; Platt 1999) has been utilized that breaks down the SVM quadratic programming optimization to simplify implementation, speed up computation, and save memory.

4 Experimental Setup and Design

All experiments were implemented in a machine with 2.20 GHZ, i3 CPU and 8 of RAM. The performance of SVM and random forest classifiers is evaluated against the test set of the modified version of the AHDBase dataset (El-Sherif and Abdelazeem 2007). More details of the modification are described later. Features have been extracted, and systems have been devolved and evaluated under the MATLAB environment. The performance of each classifier has been evaluated in terms of accuracy which is the proportion of the total number of predictions that are correct (Fawcett 2006):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (14)$$

The decision made by the classifier can be represented by a 2×2 confusion matrix including: true positives (TP) that the observations correctly labeled as positives, false positives (FP) that the observations incorrectly labeled as positive, true negatives (TN) that corresponds to negative observations correctly labeled as negative, and false negatives (FN) that refers to positive observations incorrectly labeled as negative.































Arabic Handwritten Digits database (AHDBase) has been used to investigate the performance of the proposed feature extraction techniques using sliding windows. AHDBase is a large dataset composed of 70,000 digit images, 60,000 for training, and 10,000 for testing (El-Sherif and Abdelazeem 2007). AHDBase was modified to have the same format of another large Latin Digit dataset called MINST (LeCun 1998). The Modified version was created by normalizing the height h and width w of image to have a new height h_{new} and new width w_{new} , based on the following conditions:

If $h > w$, then $h_{\text{new}} = 20$ and $w_{\text{new}} = \lfloor 20 w/h \rfloor$.

If $w > h$, then $w_{\text{new}} = 20$ and $h_{\text{new}} = \lfloor 20 h/w \rfloor$.

Table 1 shows samples of 10 digits of the modified version of AHDBase dataset.

Table 1 Samples of the modified version of AHDBase dataset

Latin digit	0	1	2	3	4	5	6	7	8	9
Arabic digit	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
MAHD Base										
										
										

The first row shows the shape of the Latin digits, and the second row shows the corresponding Arabic digits

It seems inadequate to extract the features in global level because of the similarities issues occurred due to the rotation issues of writers. Alternatively, the proposed features are extracted using the sliding window to obtain more informative features. In this study, four sizes sliding windows of 16×16 , 16×8 , 8×16 , 8×8 pixels are evaluated, all experiments have been implemented as the following; digit image was scaled into 32×32 pixels, and then a sliding window of 50% overlap was passed vertically and horizontally generating local regions.

Features were extracted from each region, and then all features of every region were concatenated producing the feature vector. Random forest and SVM classifiers were used to explore the importance of the proposed sliding windows and features. Moreover, there are some important parameters for classifiers, “number of trees” and “number of features” for random forest and the kernel function for SVM. Random forest classifier is set up with 70 trees, based on (Oshiro et al. 2012) that proved that the significant number of trees is between 64 and 128, and a subset of features determined with $(\lfloor \log_2(\# \text{ features}) + 1 \rfloor)$ (Breiman 2001).

5 Results and Discussion

This section analyzes the experiments carried out and discusses the main results attained by the different approaches used in our proposed scheme. Before discussing the results, the accuracies of all experiments are shown in Table 2. The rows in Table 2 represent the accuracies obtained using a corresponding sliding window, and the columns illustrate the feature size generated using the feature extraction technique and the sliding window, and the classifiers.

Table 2 The obtained accuracies

Feature extraction techniques	Sliding window	Feature size	SVM (linear)	SVM (RBF)	Tree
Mean-based	None	1	30.46	30.11	23.84
	16 × 16	9	81.6	84.37	86.69
	16 × 8	21	91.77	94.47	95.59
	8 × 16	21	92.2	94.66	95.24
	8 × 8	49	96.62	97.89	98
GLCM-based	None	7	45.44	46.39	44.01
	16 × 16	63	94.11	96.01	93.71
	16 × 8	147	97.43	97.45	96.68
	8 × 16	147	97.43	97.69	96.86
	8 × 8	343	98.33	83.14	98
Moment-based	None	8	28.19	28.42	31.01
	16 × 16	72	53.15	56.75	94.07
	16 × 8	168	21.43	35.67	97.61
	8 × 16	168	34.3	36.12	95.85
	8 × 8	392	21.61	25.56	98
Edge direction histogram	None	20	81.23	87.19	86.34
	16 × 16	180	99	98.92	98.66
	16 × 8	420	99.1	94.24	98.84
	8 × 16	420	99.05	92.92	98.65
	8 × 8	980	99.13	68	98.7

5.1 Significance of Sliding Windows

The effect of the sliding windows can be observed in Table 2. When the features were extracted without using sliding window, the best accuracies are 30.46%, 46.39%, 31.01% and 87.19% obtained by Mean-based features with linear-SVM, GLCM-Based with RBF-SVM, Moment-Based with the tree, and EDH features with RBF-SVM classifier, respectively. However, the accuracies are increased dramatically when sliding windows are used. All the best accuracies obtained by the proposed sliding windows are between 86.69 and 99.13% where the lowest and the highest of the best accuracies have been produced using Mean-based and EDH techniques, respectively. In addition, the highest accuracies acquired using 16 × 16, 16 × 8, 8 × 16 and 8 × 8 pixels as sliding windows are 99%, 99.1%, 99.05% and 99.13%, respectively. Moreover, the performance listed above shows that the sliding window of 8 × 8 pixels gives the highest accuracies which points out an assumption that the smaller the sliding window, the higher the accuracy. Although, the Moment-based technique with SVM classifier rejects this hypothesis, the tree classifier supports it using the same features. Based on this, it can be concluded that SVM cannot be trained a good model using the Moment-based features. Also, EDH with RBF-SVM cannot be considered for rejecting this hypothesis but linear-SVM accepts it that indicates the issue happened due to the RBF kernel itself.

5.2 Significance of Features

As sliding window approach is considered as a significant factor to enhance the recognition rate, the nature of extracting features can be considered as an important factor, too. From accuracies in Table 2, it can be noticed that even the sliding window is not used with EDH technique, the accuracy is 87.19% using RBF-SVM classifier. To simplify the exploration of the significance of the used techniques, the highest accuracies of each feature extraction technique are considered regardless of the sliding window and the classifier. The best performance obtained by Mean-based, GLCM-based, Moment-based, and EDH are 98%, 98.33, 98% and 99.13%, respectively. However, EDH gives the best recognition rate; it has the highest number of features that increases the consuming time of the processing. On the other hand, whereas Mean-based produces the lowest accuracy, the number of features used in this technique is sevenfold less, eightfold less, and 20-fold less than those used in GLCM-based, Moment-based, and EDH, respectively, implying that the strength of every Mean-based features is more significant than others.

5.3 Significance of Classifiers

Many classifier techniques were proposed in the machine learning area to address the challenging problems because one or a few of them cannot produce the satisfactory performance because of the nature of training/testing data as

well as the complexity of processing those data. In this study, two different classification algorithms have been utilized to investigate the obtained features. From Table 2, it can be noticed that the highest accuracy of 56.75% obtained by Moment-based technique using SVM classifiers is not acceptable, while tree classifier gives an accuracy of 98%. In this case, the issue is not due to the sliding windows or the feature extraction technique, but the problem is related to the compatibility between SVM and the data. In other words, this is due to the harmonization of data to some classification algorithms and the incompatibility to others.

The confusion matrix for EDH features is depicted in Table 4. The symbol RR% represents the accuracies of the corresponding digits or recognition rates. From Table 4, it can be noted that the model recognizes digits “4” and “7” higher than other digits in which the misclassification of

both digits has been occurring only with 4 samples for each. On the other hand, the lowest recognition rate happens with digit “0” where it has been recognized as digit “5” ten times and as digit “1” four times, this is because of the high convergence that happens due to the variation of the writing styles. Furthermore, the similarity between digits “2” and “3” is also high in some writing styles that makes the model recognize digit “3” as digit “2” seven times and digit “2” as digit “3” four times.

6 Applying the Proposed Work to Other Datasets

After discussing the significance of the sliding window approach, extracted features, and the classifiers applied, we have validated our study using two Farsi/Arabic digit

Table 3 A comparison between the accuracies obtained with and without using sliding window approach

Tech	Dataset	SW	Classifier	Acc
Mean-based	AHDB	None	SVM + linear	30.46
		8×8	Tree	98
	IFHCDB	None	SVM + RBF	33.34
		8×8	SVM + RBF	82.01
	HODA	None	Tree	25.63
		8×8	SVM + RBF	97.16
	MNIST	None	SVM + linear	22.54
		8×8	SVM + RBF	96.37
	GLCM-based	None	SVM + RBF	46.39
		8×8	SVM + linear	98.33
GLCM-based	IFHCDB	None	SVM + linear	28.19
		8×8	SVM + linear	49.46
	HODA	None	SVM + linear	42.07
		8×8	SVM + linear	97.33
	MNIST	None	SVM + RBF	45.70
		8×8	SVM + linear	97.36
	Moment-based	None	SVM + RBF	28.42
		8×8	Tree	98
	IFHCDB	None	SVM + RBF	28.90
		8×8	Tree	81.21
Moment-based	HODA	None	Tree	27.10
		8×8	Tree	95.86
	MNIST	None	SVM + RBF	24.81
		8×8	Tree	94.76
	Edge direction histogram	None	SVM + RBF	87.19
		8×8	SVM + linear	99.13
	IFHCDB	None	SVM + RBF	91.99
		16×16	SVM + RBF	95.72
	HODA	None	SVM + RBF	77.70
		16×16	SVM + RBF	98.66
Edge direction histogram	MNIST	None	SVM + RBF	71.08
		8×8	SVM + linear	98.77

Table 4 The confusion matrix of Edge Direction Histogram

		Predicted category											
		0	1	2	3	4	5	6	7	8	9	RR (%)	
Actual category	0	0	982	4	1	1	1	10	0	0	1	0	98.2
	1		4	994	0	1	0	0	0	0	0	1	99.4
	2		1	1	988	4	4	2	0	0	0	0	98.8
	3		0	3	7	988	0	0	0	0	1	1	98.8
	4		0	2	2	0	996	0	0	0	0	0	99.6
	5		4	0	3	0	1	990	0	1	1	0	99
	6		0	3	0	0	1	0	993	0	0	3	99.3
	7		0	1	0	0	0	2	1	996	0	0	99.6
	8		0	3	1	1	0	1	0	0	994	0	99.4
	9		0	1	1	0	1	1	4	0	0	992	99.2
Total													99.13

The bold values represent the digits that produce the highest recognition rates

datasets called IFHCDB (Mozaffari et al. 2006) and HODA (Khosravi and Kabir 2007), and one Latin digit dataset called MNIST (LeCun 1998). The number of images in training sets is 12,419, 60,000, and 60,000 and in testing sets is 5321, 20,000 for IFHCDB, HODA and MNIST, respectively.

The same experimental setup has been applied to these datasets. However, the highest accuracy acquired using the sliding window has been compared against the accuracy obtained without using the sliding window. Table 3 shows the comparison of the four feature extraction techniques using four Digit datasets, one for Arabic language, two for Farsi/Arabic digit, and one for Latin language. The Mean-based gives accuracies of 30.46%, 33.34%, 25.63% and 22.54% for AHDB, IFHCDB, HODA and MNIST, respectively, when the sliding window is not applied. However, the accuracies increase threefold to fourfold when size of 8×8 of sliding window is used. In addition, the effect of sliding window to GLCM-based and Moment-based techniques is almost the same as the effect of the Mean-based technique (Table 4).

Moreover, the impact of sliding window to EDH technique is positive to enhance the accuracy where the difference between accuracies is 3.73, 11.94, 20.96 and 27.69 for IFHCDB, AHDB, HODA and MNIST, respectively. The best sizes of sliding window to EDH are 8×8 for Arabic (AHDB) and Latin (MNIST) datasets and 16×16 for Farsi/Arabic (IFHCDB and HODA) datasets. Furthermore, it can be concluded that the features extracted using EDH are the most significant because their accuracies acquired without using the sliding window are better than those obtained using the others.

7 Comparison Against Previous Studies

In this section, a comparison with other works has been discussed. Table 5 shows the recognition rate of our proposed work compared to others. The columns in Table 5 represent, from left to right, the feature extraction approaches, sliding windows used (SW), the complexity or the domain of the data (C) which are either complex, Y, or simple, N. Then the dataset used comes after the complexity, the recognition rate (RR), and the feature size of the corresponding technique (FS).

In this comparison, the techniques that have accuracies of 98% in this proposed work are considered to compare against the studies that have been done in the literature review section. The comparison is done based on the following criteria: accuracy, feature size, complexity of the implementation or the domain, and database used. The accuracies of the literature review are sorted from lowest to highest, and the obtained accuracies of this paper are also sorted separately that simplifies the comparison.

Starting with the lowest obtained accuracies using Mean-based and Moment-based techniques that are implemented simply, these approaches produce higher accuracy of 98% than that acquired in AlKhateeb and Alseid (2014), Radwan (2013), Takruri et al. (2015) and Shilbayeh et al. (2013). However, they have feature size larger than them except (Takruri et al. 2015) that uses 400 features. In addition, the features of AlKhateeb and Alseid (2014) and Radwan (2013) are extracted after applying the transform approach.

GLCM-Based gives a better recognition rate than recognition rates of Ghaleb et al. (2013), but features extracted in Ghaleb et al. (2013) are not complex as GLCM-Based. However, both GLCM-Based features and that of Lawgali (2015) are complex and give almost the

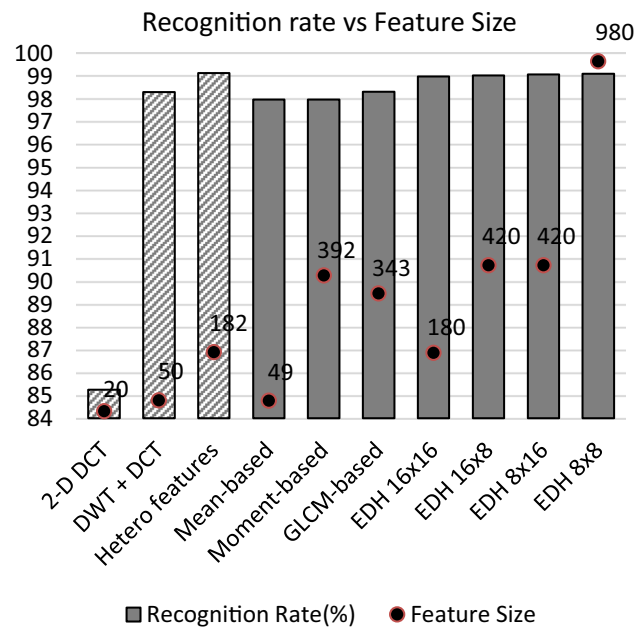
Table 5 The comparison with other studies

References	Feature extraction technique	SW	C	Datasets	RR (%)	FS
Shilbayeh et al. (2013)	Geometric-based	N	N	Private dataset	74.4	18
AlKhateeb and Alseid (2014)	2D DCT	N	Y	AHDBase	85.26	20
Takruri et al. (2015)	Pixel intensity	N	N	CENPARMI	89	400
Ghaleb et al. (2013)	Hetero-features	N	N	Private dataset	98.15	NA
Lawgali (2015)	DWT + DCT	N	Y	AHDBase	98.32	50
Rashnodi et al. (2011)	DFT + geometrical-based	N	Y	Persian numerals	98.84	128
Mahmoud and Al-Khatib (2011)	Gabor-based	Y	Y	CENPARMI	98.95	432
Awaida and Mahmoud (2014)	GSC	N	N	CENPARMI	99.04	288
El-Sherif and Abdelazeem (2007)	Hetero-features	N	N	AHDBase	99.15	182
Radwan (2013)	Hierarchical features	N	Y	IFHCBD	93	5
Our work	EDH	16 × 16	Y	IFHCBD	95.72	180
	Mean-based	Y	N	AHDBase	98	49
	Moment-based	Y	N	AHDBase	98	392
	GLCM-based	Y	Y	AHDBase	98.33	343
	EDH	16 × 16	Y	AHDBase	99	180
		8 × 16			99.05	420
		16 × 8			99.1	420
		8 × 8			99.13	980

same accuracy; the number of features used in GLCM-Based is sixfold more than those extracted in Lawgali.

Moreover, EDH produces the best performance among the proposed feature extraction techniques, while its implementation is complex. EDH techniques applied using one of the proposed sliding windows give recognition rate higher than Rashnodi et al. (2011) and Mahmoud and Al-Khatib (2011). However, the feature size in Rashnodi et al. (2011) is less than them, and the number of features used in Mahmoud and Al-Khatib (2011) is close to those extracted by EDH using sliding windows of 16×8 and 8×16 . Moreover, when EDH is compared to techniques applied in Awaida and Mahmoud (2014) and El-Sherif and Abdelazeem (2007), it can be concluded that the performance difference is no more than 0.15% which represents only 15 samples of 10,000 samples of the test set data which is just not significant. Nevertheless, the complexity of Awaida and Mahmoud (2014) and El-Sherif and Abdelazeem (2007) is lower than EDH. Moreover, when EDH has been applied to IFHCBD dataset using sliding window of 16×16 pixels, the acquired accuracy is better than the accuracy of Radwan (2013), while the number of features is higher.

In addition, Fig. 2 depicts a comparison between the proposed approaches against the techniques that used the same dataset, namely AHDBase, in terms of recognition rate and feature size. It is clear that by using the same dataset, the recognition rates are significant compared to the others (of dark upward diagonal rectangles).

**Fig. 2** Comparison against previous studies that used AHDBase as a dataset

Furthermore, the proposed work also has been compared against one of the Android mobile applications proposed by McIntosh et al. (2019). They evaluated their performance using MNIST dataset. The *complex* term used in this paper, the time complexity, feature size, and the accuracy have been used as factors in this part of comparison.

Both our and McIntosh et al. techniques use none-complex implementation of features, but the feature size of ours (343 features) is less than of McIntosh et al. (784 features). In the study of McIntosh et al. (2019), the highest accuracy of 95.66% was produced using multilayer perceptron network (MLP), while the highest accuracy of this study is 97.36% when SVM is used as a classifier and SMO is used as optimization approach.

The time complexity of SVM is from $O(n)$ up to $O(n^{2.2})$ for n training instances (Platt 1999), and for classification time is $O(c)$ per instance (Yang et al. 2003). However, the time complexity of MLP for training is $O(nabc)$, where a is input neurons, b hidden neurons, and c output neurons, and for classification set $O(a + b + c)$ (Mizutani and Dreyfus 2001).

8 Comparison Against a Deep Learning Approach

Local Binary Convolutional Neural Network (LBCNN) (Juefei-Xu et al. 2016) is a variation of Convolutional Neural Networks (CNN) (LeCun et al. 1998), which uses LBP technique (Ojala et al. 1994) as a Local Binary Convolution (LBC). The significant difference between the LBC and CNN is that LBC has less number of learnable parameters than CNN. We have used the same architecture of LBCNN used in Juefei-Xu et al. (2016) with 3×3 filter size. The experiments have been run using 30 LBC units which are equivalent to 60 convolutional layers, 512 randomly generated anchor weights (LBC), 128 hidden units in the fully connected layer, 16 output channels, and 0.9 of sparsity level, which refers to the percentage of nonzero elements.

The accuracies of applying LBCNN to AHDBase, MNIST, HODA, and IFHCBD are 98.96%, 99.51%, 98.37% and 98.72%, respectively, while the accuracies obtained by EDH are 99.13%, 98.77%, 98.66% and 95.72%, respectively. It is clear to note that the performance of LBCNN with MNIST and IFHCDB datasets is better than EHD, but EDH gives the highest accuracies for AHDBase and HODA datasets.

9 Conclusion and Future Work

In this study, sliding windows are used to enhance recognition rates investigated using random forests (RF) and support vector machine (SVM) classifiers to Arabic digit images. We discussed three significant points, including the importance of sliding windows, the size of the sliding windows that are passed for extracting features, and the

importance of feature extraction techniques. To this end, four feature extraction techniques inclusive, Mean-based, GLCM-based, Moment-based, and edge direction histogram are applied to Arabic digit image. The experiments show the significance of using sliding windows either by extracting features from spatial domain or frequency domain. Moreover, the effectiveness of feature extraction techniques and the classifier algorithms are explored. The highest recognition rates acquired using the modified version of AHDBase dataset are 98% when Mean-based and Moment-based are used with RF classifier, and 98.33% and 99.13% when GLCM and EDH are applied with linear-kernel SVM, respectively.

Based on the efficiency of the machine, it can be concluded as follows. If a device of small processor such as smart mobile will use a digit recognition system, the spatial domain based on feature extraction techniques is proposed as Mean-based or Moment-based features that gives an accuracy of 98% which is good enough and acceptable. However, Mean-based technique using sliding window of 8×8 pixels produces 50 features that are almost eightfold less than Moment-based and uses a tree-based classifier, which is consider one of the best real-time classifiers, resulting in less processing time. On the other hand, if the recognition rate is more important than the efficiency, EDH as a feature extraction technique with using sliding window of 8×8 pixels and linear-kernel SVM is better than others. Furthermore, the acquired accuracies using the proposed systems are higher than or comparable against the state-of-the-art digit recognition systems.

There are many adaptations, tests, and experiments that have been left for the future because of lack of hardware resources such as RAM. For time, as the feature size increases, the time of training models increases, and it needs bigger size of RAM. Because of these reasons, some restriction has been enforced in all experiments such as only image size of 32×32 pixels has been used to decrease the number of features, while the proposed methods should be validated using different image sizes. Moreover, only four sliding window sizes have been applied for the same reasons, while more window sizes have to be tested. On the other hand, the proposed systems have been developed using only two classifiers and validated by one dataset that may be a biased dataset to the proposed systems, so it is hard to generalize this to other datasets.

Acknowledgements The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Education Malaysia for financially supporting this research under the Fundamental Research Grant Scheme (FRGS), Vote No. 1641.



References

- AlKhateeb JH, Alseid M (2014) DBN-Based learning for Arabic handwritten digit recognition using DCT features. In: 2014 6th international conference on computer science and information technology (CSIT). IEEE, pp 222–226
- Awaida SM, Mahmoud SA (2014) Automatic check digits recognition for Arabic using multi-scale features, HMM and SVM classifiers. *Br J Math Comput Sci* 4:2521
- Bosch A, Zisserman A, Munoz X (2007) Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on image and video retrieval. ACM, pp 401–408
- Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory. ACM, pp 144–152
- Breiman L (2001) Random forests. *Mach Learn* 45:5–32
- El-Sherif EA, Abdelazeem S (2007) A two-stage system for Arabic handwritten digit recognition tested on a new large database. In: Artificial intelligence and pattern recognition. pp 237–242
- Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognit Lett* 27:861–874
- Flusser J, Suk T (2006) Rotation moment invariants for recognition of symmetric objects. *IEEE Trans Image Process* 15:3784–3790
- Gao X, Xiao B, Tao D, Li X (2008) Image categorization: graph edit distance + edge direction histogram. *Pattern Recognit* 41(10):3179–3191
- Ghaleb MH, George LE, Mohammed FG (2013) Numeral handwritten Hindi/Arabic numeric recognition method. *Int J Sci Eng Res* 4(1):229–518
- Gonzalez RC, Woods RE (2008) Digital image processing, 1st edn. Pearson Prentice Hall, Upper Saddle River, NJ
- Haghighat M, Zonouz S, Abdel-Mottaleb M (2013) Identification using encrypted biometrics. In: Computer analysis of images and patterns. Springer, pp 440–448
- Haralick RM, Shanmugam K et al (1973) Textural features for image classification. *IEEE Trans Syst Man Cybern* 6:610–621
- Hu M-K (1962) Visual pattern recognition by moment invariants. *IRE Trans Inf Theory* 8:179–187
- Juefei-Xu F, Boddeti VN, Savvides M (2016) Local binary convolutional neural networks. *ArXiv Prepr. ArXiv: 160806049*
- Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK (2001) Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Comput* 13:637–649
- Khosravi H, Kabir E (2007) Introducing a very large dataset of handwritten Farsi digits and a study on their varieties. *Pattern Recognit Lett* 28:1133–1141
- Lawgali A (2015) Handwritten digit recognition based on DWT and DCT. *Int J Database Theory Appl* 8:215–222
- LeCun Y (1998) The MNIST database of handwritten digits. [Httpyann Lecun Comexdbmnist](http://yann.lecun.com/exdb/mnist)
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86:2278–2324
- Mahmoud SA, Al-Khatib WG (2011) Recognition of Arabic (Indian) bank check digits using log-gabor filters. *Appl Intell* 35:445–456
- McIntosh A, Hassan S, Hindle A (2019) What can Android mobile app developers do about the energy consumption of machine learning? *Empir Softw Eng* 24(2):562–601
- Mizutani E, Dreyfus SE (2001) On complexity analysis of supervised MLP-learning for algorithmic comparisons. In: International joint conference on neural networks, 2001, IJCNN'01. Proceedings. IEEE, pp 347–352
- Mozaffari S, Faez K, Faradji F, Ziaratban M, Golzan SM (2006) A comprehensive isolated Farsi/Arabic character database for handwritten OCR research. In: Tenth international workshop on frontiers in handwriting recognition. Suvisoft
- Ojala T, Pietikainen M, Harwood D (1994) Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Proceedings of the 12th IAPR international conference on pattern recognition, vol 1-conference a: computer vision and image processing. IEEE, pp 582–585
- Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int J Comput Vis* 42:145–175
- Oshiro TM, Perez PS, Baranauskas JA (2012) How many trees in a random forest? In: *MLDM*. Springer, pp 154–168
- Partio M, Cramariuc B, Gabbouj M, Visa A (2002) Rock texture retrieval using gray level co-occurrence matrix. In: Proceedings of 5th nordic signal processing symposium. Citeseer
- Plamondon R, Srihari SN (2000) Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans Pattern Anal Mach Intell* 22:63–84
- Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Schölkopf B, Burges CJC, Smola AJ (eds) *Advances in Kernel methods—support vector learning*. MIT Press, Cambridge, pp 185–208
- Radwan E (2013) Hybrid of rough neural networks for Arabic/Farsi handwriting recognition. *Int J Adv Res Artif Intell* 2:39–47
- Rashnodi O, Sajedi H, Abadeh MS (2011) Using box approach in persian handwritten digits recognition. *Int J Comput Appl* 32(3)
- Shilbayeh NF, Aqel MM, Alkhateeb R (2013) Recognition offline handwritten Hindi digits using multilayer Perceptron neural networks. *World Sci Eng Acad Soc* 94–103
- Takruri M, Al-Hmouz R, Al-Hmouz A, Momani M (2015) Fuzzy C means based hybrid classifiers for offline recognition of handwritten Indian (Arabic) numerals. *Int J Appl Eng Res* 10:1911–1924
- Trier ØD, Jain AK, Taxt T (1996) Feature extraction methods for character recognition—a survey. *Pattern Recognit* 29:641–662
- Vapnik VN, Chervonenkis AY (1971) On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab Appl* 16:264–280
- Yang Y, Zhang J, Kisiel B (2003) A scalability analysis of classifiers in text categorization. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 96–103