Journal of Imaging Science and Technology® 63(2): 000000-1–000000-10, 2019.

Multi-Language Handwritten Digits Recognition based on Novel Structural Features

Jaafar M. Alghazo
College of Computer Engineering and Sciences, Prince Mohammad Bin Fahd University, Saudi Arabia

Ghazanfar Latif
College of Computer Engineering and Sciences, Prince Mohammad Bin Fahd University, Saudi Arabia
Faculty of Computer Science and Information Technology, University of Malaysia, Sarawak, Malaysia
E-mail: glatif@pmu.edu.sa

Loay Alzubaidi
College of Computer Engineering and Sciences, Prince Mohammad Bin Fahd University, Saudi Arabia

Ammar Elhassan
King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Jordan

Abstract. *Automated handwritten script recognition is an important task for several applications. In this article, a multi-language handwritten numeral recognition system is proposed using novel structural features. A total of 65 local structural features are extracted and several classifiers are used for testing numeral recognition. Random Forest was found to achieve the best results with an average recognition of 96.73%. The proposed method is tested on six different popular languages, including Arabic Western, Arabic Eastern, Persian, Urdu, Devanagari, and Bangla. In recent studies, single language digits or multiple languages with digits that resemble each other are targeted. In this study, the digits in the languages chosen do not resemble each other. Yet using the novel feature extraction method a high recognition accuracy rate is achieved. Experiments are performed on well-known available datasets of each language. A dataset for Urdu language is also developed in this study and introduced as PMU-UD. Results indicate that the proposed method gives high recognition accuracy as compared to other methods. Low error rates and low confusion rates were also observed using the novel method proposed in this study.*

[DOI: 10.2352/J.ImagingSci.Technol.2019.63.2.000000]

1. INTRODUCTION

Handwritten scripts are intricate with several factors affecting their complexity, including writer-specific variations subject to inter-writer and intra-writer variables. This applies to all handwritten scripts including numerals. Several recognition algorithms have been developed for offline handwritten recognition for various applications [1, 2]. With the increase in popularity and type variation of gesture, touchscreen, and handheld devices, the need for novel algorithms to detect and automatically recognize handwritten/gestured numerals becomes a significant requirement that determines success of these devices [3, 4]. The number of applications that depend on accurate, automatic online recognition of handwritten numerals is increasing. Applications vary from teaching children to write numerals to secure banking sector. In all cases, a robust and accurate handwritten numeral recognition system is needed. Rather than compile a list of applications that require this feature, we list some of the salient industries that depend on this technology:

1. Banking and Finance
2. Education at all levels from K-12 to Higher Education
3. Supply Chain Management (all variations including delivery)
4. Food and Restaurant Industry
5. In-Vehicle Navigation, Media and Entertainment.

These application areas constitute part of the motivation for the work presented in this paper. Conventional input devices and keyboards are available alongside stylus, gesture, and finger scripting in most devices. The average user needs or prefers to use the gesture, stylus, or fingertip mode with several applications. As a result, gradually these devices are dropping support for conventional input. With a universal service-based, online handwritten recognition system, the need for customizing different applications for online digit recognition will no longer be an issue, thereby cutting costs and reducing the time to market. Furthermore, universal support allows many current software applications to be retro-upgraded such that they now cater to the needs of larger user bases. Point of Sale (POS) devices will be usable in much larger geographical areas without additional modifications, Intra-bank systems become more compatible, thus aiding in security [5, 6]. There are many more advantages that can be listed for having a universal handwritten numeral recognition system and this partial list is sufficient to justify and motivate this unique approach.

Received Jan. 24, 2018; accepted for publication Oct. 22, 2018; published online xx xx, xxxx. Associate Editor: Yeong-Ho Ha.

1062-3701/2019/63(2)/000000/10/$25.00

Whether it is online automatic numeral recognition or offline numeral recognition, most previous research has concentrated on developing algorithms for the recognition of numerals in one language or couple of languages with digit similarity and resemblance. In this paper, we develop a novel Universal Recognition Algorithm that is able to achieve high automatic recognition accuracy rates for numerals with support for at least six languages. The proposed automatic numerals recognition system supports digit recognition in English, Arabic, Persian, Urdu, Devanagari (Marathi) and Bengali, which in total, are spoken by approximately 1.86 billion people worldwide. The main contribution is an algorithm of unique enhanced structural features that achieves the ultimate goal of numeral recognition in multiple languages. To the authors' knowledge, this is the first time that a system is developed for a unified recognition of multiple languages. Although its recognition rate may be comparable with rates reported in the extant literature, an important distinction is that the latter applies to one or two languages maximum. The extracted 65 features were used as input to various classifiers, including Artificial Immune, Multi-Layer Perceptron (MLP), Logistic and Random Forest (RF). The latter classifier was found to achieve the best accuracy. The system was tested on six well-known online available datasets for six different languages with an overall accuracy average of 96.73%. Additionally, we combined three of the most similar languages (Arabic Eastern, Persian, Urdu) datasets into one large dataset of 158,500 instances with 126,500 training instances and 32,000 instances for testing. The system with combined dataset was able to achieve an average of 97.26% accuracy.

This article is organized as follows: section 2 presents a review of the literature related to the proposed work. Section 3 describes the methodology and various processes of the proposed approach while section 4 discusses the structure of the experimental database. Section 5 presents results obtained from numeral recognition system testing and section 6 contains conclusion and future work.

2. LITERATURE REVIEW

As indicated above, most previous research was aimed at developing methodologies for the recognition of numerals in a single language while some extended to two or three languages with the caveat of having similar digits with high resemblance rates. Alkhateeb and Alseid [7] proposed a Dynamic Bayesian Network (DBN) based system for Eastern Arabic handwritten digit recognition using Discrete Cosine Transform (DCT) features. They tested their system on the Arabic Handwritten database (ADBase) consisting of 70,000 records written by 700 writers and achieved an average of 85.26% recognition rate accuracy. Salimi and Giveki [8] proposed an algorithm based on a group of Singular Value Decomposition (SVD) classifiers and multiphase Particle Swarm Optimization (PSO) for the recognition of Farsi/Arabic handwritten digits. They tested their proposed system on the HODA database achieving accuracy rates of 97.02%. Musleh, Halawani and Mahmoud [9] proposed an algorithm based on fuzzy logic for handwritten Arabic digit recognition. The classification in their proposed algorithm is done over two phases: (i) zero/nonzero classification using Support Vector Machine (SVM) classifier and (ii) classification of 1–9 digits using a syntactic fuzzy classifier. Tests of this system were applied to a database of 32695 digits. This system achieved an accuracy rate of 99.55% for the zero/nonzero classification phase and 98.01% for the 1–9 digits classification phase. It should be noted here that by classifying the numbers into zero/nonzero, the system automatically increases in accuracy because of the confusion between 0 and other digits, like 5 written in Arabic. Hosseinzadeh, Razzazi and Kabir [10] proposed a novel, large-margin domain adaptation method for the recognition of isolated handwritten digits. They also developed a framework of ensemble projection feature learning to use the available unlabeled samples in the target domain. This approach was tested on three standard datasets namely: MNIST (Western Arabic), USPS, and ICDAR and showed that their proposed architecture performs better than several known domain adaptation methods, both in supervised and semi-supervised domain adaptation scenarios with mean accuracy of 89.86%.

Sadri, Yagenehzad and Saghi [11] constructed a fairly comprehensive dataset for offline Persian handwritten digital character recognition. The dataset contains 97124 digits in addition to other format data, including writer's age and gender, worded dates, numeral dates, and numeral strings collected from 500 Persian native speakers. Boukharouba and Bennia [12] proposed a feature extraction technique for recognition of handwritten Persian digits based on transition information in the vertical and horizontal directions of the image grouped with the chain code histogram (CCH). They utilized Support Vector Machine (SVM) classification method and tested their approach against the HODA dataset achieving an average accuracy of 98.55%. Karimi et al. [13] proposed a Persian handwritten digit recognition approach based on 115 extracted features in combination with the ensemble classifiers, which was tested on the TMU database and achieved accuracy rates of 92.8%.

Sarkhel et al. [14] proposed a region sampling technique based on multi-objective evolutionary algorithms. They used a Non-dominated Sorting Harmony-Search Algorithm (NSHA) and a Non-dominated Sorting Genetic Algorithm-II (NSGA-II) for region sampling separately. Their approach subsequently selected the most informative set of local regions using the framework of Axiomatic Fuzzy Set (AFS) theory from the output produced by the NSHA and NSGA-II. This system was tested on two Bengali handwritten datasets with SVM as the choice for classifier. Testing relied on feature sets containing convex hull and Center of Gravity (CG) based quad-tree partitioned longest-run algorithm. The tests resulted in recognition accuracy rates close to 98%.

Basu et al. [15] proposed a method for the recognition of Bengali digits using the Dempster–Shafer (DS) technique that combines classifications obtained from Multi-Layer Perceptron (MLP) classifiers and two distinct feature sets.

Table I. Numerals in six different languages

Arabic (Western)	Arabic (Eastern)	Persian	Urdu	Bengali	Devanagari
0	٠	۰	۰	০	०
1	١	۱	۱	১	१
2	٢	۲	۲	২	२
3	٣	۳	۳	৩	३
4	٤	۴	۴	৪	४
5	٥	۵	۵	৫	५
6	٦	۶	۶	৬	६
7	٧	۷	۷	৭	७
8	٨	۸	۸	৮	८
9	٩	۹	۹	৯	९

This system was tested on a dataset of 6,000 handwritten digits with an obtained average accuracy rate of 95.1%. Wang et al. [16] proposed a hardware implementation of a Neural Network for the recognition of handwritten digits using Resistance Random Access Memory (RRAM) as synaptic weight elements. Their proposed system was tested on the MNIST (Western Arabic) dataset and achieved an average accuracy of 81%. Ali and Ghani [17] proposed a method using transformation-based features in conjunction with the Discrete Cosine Transform (2D-DCT). They applied Hidden Markov Models (HMMs) as the classifier and tested their approach against the MNIST (Western Arabic) dataset. The accuracy rate obtained from these tests exceeded 97.2%. Jie et al. [18] proposed a numeral recognition method based on the Enhanced Label Propagation algorithm in conjunction with Entropy based features to weigh the confidence coefficient of each numeral. The new confidence coefficients are fed back to the label propagation algorithm to retrain the system based on the new coefficients. The method was tested using the MNIST (Western Arabic) dataset and achieved a 98% recognition rate. Bajaj, Dey and Chaudhury [19] proposed three types of features for Devanagari numeral recognition: Density, Moment features of right, left, upper and lower profile curves, and Descriptive Moment features. Multiple classifiers are then used and connected using the meta-pi network to obtain optimum recognition rates. This system achieved around 90% accuracy rates.

3. PROPOSED METHOD

The proposed method includes the three main phases of numeral recognition systems. The first phase consists of what is referred as 'preprocessing phase' which includes segmentation, binarization, noise removal, size, and slope normalization. The second phase is the feature extraction. We propose a novel feature extraction method based on 65 local features. These 65 features, explained in detail later, are the basis for a multi-language handwritten numeral recognition system. The third and final phase consists of applying a classification technique to recognize the numerals. In this paper, we applied four different classification techniques: Artificial Immune, Multi-layer Perceptron, Logistic and Random Forest. The Random Forest classifier was found to achieve the best recognition rate for multi-language recognition with an average accuracy of 96.7%.

3.1 *Preprocessing*

As stated above, the variations of handwritten numerals are endless with several factors affecting these variations. The recognition of handwritten numerals is a complex problem due to the variations of the handwriting which is further increased in complexity when trying to develop an algorithm to recognize numerals in multiple languages that are not closely related together. Table I shows the writing style of numerals in the six different languages we are targeting in this work. As noticed in Table I, some languages have numerals whose writing configurations are closely related together, such as Arabic and Persian. The numerals in these two languages are the same with minor differences in the numbers 4, 5, and 6. Earlier work in the literature has targeted the development of algorithms in more than one language whose numerals have similarities. No research has been done on unified numeral recognition systems for multiple languages with different styles of numerals.

Handwritten writing styles vary tremendously depending on inter-writer and intra-writer variables among others. The research shows that 52 writing classes with minor variations exist for Arabic and Persian numerals only [20] and other languages are expected to have a similar number of writing classes if not more.

The numeral images must be preprocessed to make them ready for the other two phases in the recognition process. Preprocessing includes many variables, such as shape of the image, location of the numeral in the image, size of the numeral in the image, angle at which the numeral was written, and noise. In the preprocessing phase, first the image is segmented to separate each digit based on the boundary box. The digits are then centered in the middle of the boundary box and rotated (if needed) as well. Each digit is then normalized to a 28 × 28 pixel image. The numerals are then subjected to a binarization process from grayscale by using Otsu's thresholding method [21]. Finally, noise removal from the binarized image is executed using morphological operations, such as dilation and erosion, and a 3 × 3 window disk shaped structure. It should be noted here that some existing online datasets are available with the preprocessing already done. However, some datasets supply only the raw data to which preprocessing must be applied. We have developed the Urdu dataset used for this study and have applied the preprocessing phase to that as well.

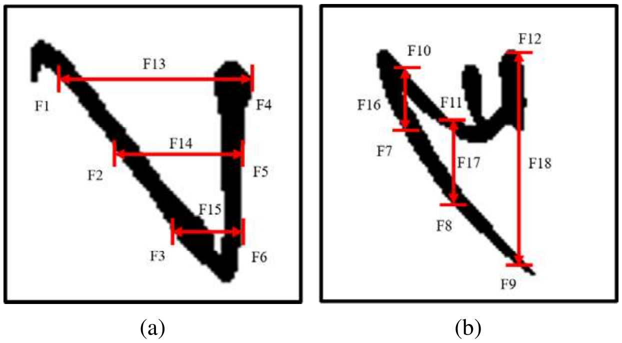
3.2 *Feature Extraction*

Feature extraction is an important and vital part of the digit recognition process. In fact, the main contribution of this work lies in developing a novel feature extraction method which is detailed below. Similarities in the writing structure of some digits make the local structural features of the digits more important than the global features, such as DCT, DFT and Histogram based features [22, 23]. Our novel feature extraction method is composed of a total of 65 local structural features extracted as detailed below. Since numerals in all languages are written in a specific shape and style and recognized through observation of the individual, the supervised learning approach proposed in this paper considers the structural features of each numeral in the targeted languages. The features chosen are based on a comprehensive analysis by the authors of the shape and structure of numerals in the chosen languages. Based on the observations of the authors, different features detailed below provide sufficient information to the system for the differentiation and classification of the numerals in different languages. The choice of these features was also based on a thorough literature review by the authors on different feature extraction methods for recognition of both numeral, letters, and text in different languages. The way in which humans view and differentiate numerals in different languages was translated into features that could be measured mathematically to allow for machine learning, classification, and recognition.

1. Let us suppose that the preprocessed numeral binary image is B with equal width n and height m, represented as a square matrix $b\left(x, y\right)$.
2. B is divided horizontally from left to right as shown in Figure 1(a) and horizontally from up to down as shown in Fig. 1(b). Three starting and three ending x-axis values based on the occurrence of black pixels are calculated. Similarly, three starting and three ending y-axis values are calculated. Both horizontal and vertical measurements give 12 features. Equations (1)–(4) represent the calculations for these features.

$$F_{i\leftarrow 1\,to\,3} = \lim_{k\leftarrow\frac{n}{4}\times i}\left[\lim\nolimits_{l\leftarrow 1:n}\left(\min\nolimits_{b(x_l,y_k)\rightarrow 1} x_l\right)\right] \tag{1}$$

$$F_{i\leftarrow 4\,to\,6} = \lim_{k\leftarrow\frac{n}{4}\times i}\left[\lim\nolimits_{m\leftarrow 1:n}\left(\max\nolimits_{b(x_l,y_k)\rightarrow 1} x_l\right)\right] \tag{2}$$



(a) (b)

Figure 1. Proposed horizontal and vertical features extraction (digit samples from Arabic). From left to right: (a). Three starting (F1, F2, F3) and three ending x-axis (F4, F5, F6) values based on black pixels, and distance between first occurrence and last occurrence of black pixel (F13, F14, F15). (b). Three starting (F7, F8, F9) and three ending (F10, F11, F12) y-axis values, and distance between first occurrence and last occurrence of black pixel (F16, F17, F18).

$$F_{j\leftarrow 7\,to\,9} = \lim_{k\leftarrow\frac{n}{4}\times i}\left[\lim\nolimits_{n\leftarrow 1:n}\left(\min\nolimits_{b(x_k,y_i)\rightarrow 1} y_l\right)\right] \tag{3}$$

$$F_{j\leftarrow 10\,to\,12} = \lim_{k\leftarrow\frac{n}{4}\times i}\left[\lim\nolimits_{o\leftarrow 1:n}\left(\max\nolimits_{b(x_k,y_i)\rightarrow 1} y_l\right)\right]. \tag{4}$$

3. The distance between the first occurrence and last occurrence of a black pixel is measured horizontally and vertically as represented by $F_{i\leftarrow 13\ to\ 18}$ in Fig. 1. This gives another three horizontal features and three vertical features of handwritten numeral as shown in Eqs. (5)–(6).

$$F_{i\leftarrow 13\,to\,15} = F_{i-9} - F_{i-12} \tag{5}$$

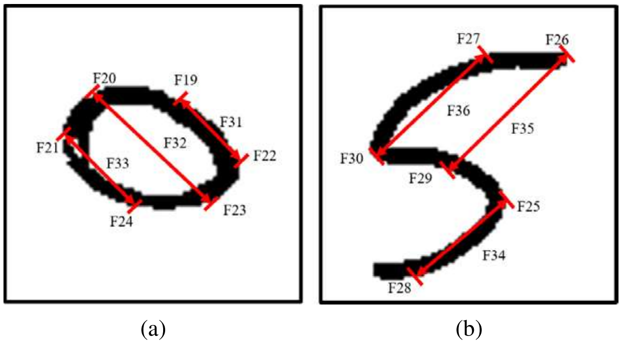
$$F_{i\leftarrow 16\,to\,18} = F_{i-6} - F_{i-9}. \tag{6}$$

4. Diagonal features are also measured based on the starting diagonal point and end diagonal point. Three starting and three ending diagonal coordinates of black pixels from top left to bottom right are calculated. Similarly, another three starting and three ending diagonal coordinates of black pixel from top right to bottom left are used as features. This adds another 12 features as indicated in Eqs. (7)–(10).

$$F_{i\leftarrow 19\,to\,21} = \begin{cases} \lim_{l,k\leftarrow s:n,1:n-s}\left(\min_{b(x_l,y_l)\rightarrow 1} x_l\right), & i = 19 \\ \lim_{l,k\leftarrow 1:n,1:n}\left(\min_{b(x_l,y_l)\rightarrow 1} x_l\right), & i = 20 \\ \lim_{l,k\leftarrow 1:n-s,s:n}\left(\min_{b(x_l,y_l)\rightarrow 1} x_l\right), & i = 21 \end{cases} \tag{7}$$

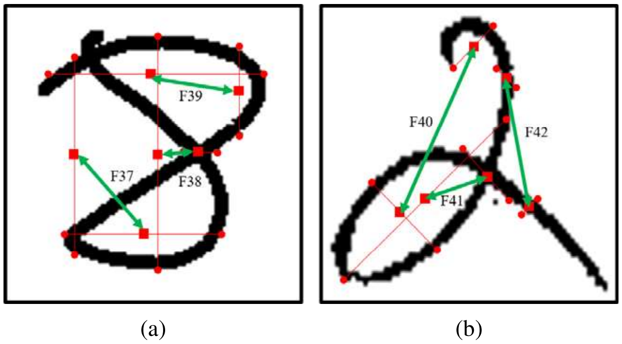
$$F_{i\leftarrow 22\,to\,24} = \begin{cases} \lim_{l,k\leftarrow s:n,1:n-s}\left(\max_{b(x_l,y_l)\rightarrow 1} x_l\right), & i = 22 \\ \lim_{l,k\leftarrow 1:n,1:n}\left(\max_{b(x_l,y_l)\rightarrow 1} x_l\right), & i = 23 \\ \lim_{l,k\leftarrow 1:n-s,s:n}\left(\max_{b(x_l,y_l)\rightarrow 1} x_l\right), & i = 24 \end{cases} \tag{8}$$

$$F_{i\leftarrow 25\,to\,27} = \begin{cases} \lim_{l,k\leftarrow s:n,1:n-s}(\min_{b^T(x_l,y_l)\rightarrow 1} y_l), & i = 25 \\ \lim_{l,k\leftarrow 1:n,1:n}(\min_{b^T(x_l,y_l)\rightarrow 1} y_l), & i = 26 \\ \lim_{l,k\leftarrow 1:n-s,s:n}(\min_{b^T(x_l,y_l)\rightarrow 1} y_l), & i = 27 \end{cases} \tag{9}$$



(a) (b)

Figure 2. Proposed diagonal features extraction (sample numerals from Persian and English). From left to right: (a). Three starting (F19, F20, F21) and three ending x-axis (F22, F23, F24) values based on black pixels, and distance between first occurrence and last occurrence of black pixel (F31, F32, F33). (b). Three starting (F25, F26, F27) and three ending (F28, F29, F30) y-axis values and distance between first occurrence and last occurrence of black pixel (F34, F35, F36).



(a) (b)

Figure 3. Height distance to width distance aspect ratio from right to left: (a): Aspect ratio against the centers of x-axis and y-axis distances, (b): Aspect ratio against the centers of left and right diagonal distances.

$$F_{i\leftarrow 28\,to\,30} = \begin{cases} \lim_{l,k\leftarrow s:n,1:n-s}(\max_{b^T(x_l,y_l)\rightarrow 1} y_l), & i=28 \\ \lim_{l,k\leftarrow 1:n,1:n}(\max_{b^T(x_l,y_l)\rightarrow 1} y_l), & i=29 \\ \lim_{l,k\leftarrow 1:n-s,s:n}(\max_{b^T(x_l,y_l)\rightarrow 1} y_l), & i=30 \end{cases} \qquad (10)$$

5. The distance between the first occurrence and last occurrence of a black pixel is measured for each diagonal of B and B transpose B^T as represented by $F_{i\leftarrow 31\,to\,36}$ in Figure 2. This gives another six diagonal distance features of handwritten numeral as shown in Eqs. (11)–(12).

$$F_{i\leftarrow 31\,to\,33} = b(xi, yi) - b(xj, yj), \qquad (11)$$

where xi, yi are the ending coordinates and xj, yj are the starting coordinates

$$F_{i\leftarrow 34\,to\,36} = b(xl, yl) - b(xm, ym), \qquad (12)$$

where xl, yl are the ending coordinates and xm, ym are the starting coordinates.

6. Height distance to width distance aspect ratios are measured based on the calculated distances horizontally and vertically as shown in Figure 3 utilizing Figs. 1 and 2. Similarly, distance to width aspect ratios are measured for the diagonals. This gives another six features, $F_{i\leftarrow 37\,to\,42}$, as shown in Eqs. (13)–(14).

$$F_{i\leftarrow 37\,to\,39} = \left|\frac{F_{i-24} - F_{i-21}}{2}\right| \qquad (13)$$

$$F_{i\leftarrow 40\,to\,42} = \left|\frac{F_{i-9} - F_{i-6}}{2}\right|. \qquad (14)$$

7. Number of black pixels of all six diagonals are calculated and used as features $F_{i\leftarrow 43\,to\,48}$ as shown in Eqs. (15)–(16).

$$F_{i\leftarrow 43\,to\,45} = \sum_{l=1}^{n} b_i(x_l, x_l) = 1 \qquad (15)$$

$$F_{i\leftarrow 46\,to\,48} = \sum_{l=1}^{n} b^T{}_i(y_l, y_l) = 1. \qquad (16)$$

8. The handwritten numeral is then divided into 4 × 4 segments, with a total 16 blocks. The number of black pixels in each block is then calculated using Eq. (18). Q.3

$$F_{i\leftarrow 49\,to\,64} = \lim_{l\leftarrow 1\times(i-49):4\times(i-49)} \left(\lim_{k\leftarrow 1\times(i-49):4\times(i-49)} \left(\sum b[x_l, y_k] = 1\right)\right). \qquad (18)$$

9. The sum of black pixels from the complete binary numeral image is calculated and used as 65th feature of the numerals using Eq. (19).

$$F_{i\leftarrow 65} = \sum_{l,\,k\leftarrow 1,1}^{n,n} b[x_l, y_k] = 1. \qquad (19)$$

3.3 *Classification*

After the feature extracting phase the numerals are input to several well-known and widely used classifiers. In this paper, we apply the Artificial Immune, Multi-layer Perceptron, Logistic, and Random Forest classifiers for experimental results. Artificial Immune Recognition System (AIRS) is a classifier that is inspired by the biological immune system [24]. The system relies on three processes: negative selection, clonal selection, and immune network. The set of data for pattern recognition is put through the three processes with multiple iteration in some processes until a predefined criterion is met for the pattern recognition to complete. MLP is a feedforward neural network that requires mapping a set of input data onto a set of outputs. It consists of three or more layers, with three being the minimum: input layer, output layer and a hidden layer [25]. Logistic regression is a third classification technique used in this study [26]. There are four classification techniques that were applied and since Random Forest [27] produced the best accuracy, we chose to briefly describe the three above techniques.

Random Forest (RF) relies on decision trees and is viewed as a grouping of several decision trees. Initially, in the training phase, each tree in the ensemble trains the system based on randomly sampled data with replacement from the training vector. A model-averaging scheme, known as Bootstrap aggregating (Bagging), is used for averaging to increase the correlation and avoid the issue of overfitting [28]. Important information about individual features can be gathered after model creation. For testing purposes, data are presented to the individual trees for classification. Votes are being collected from each individual tree, and finally the RF classifies desired output based on majority votes which is further described below.

Suppose, $x^{'}$ being the features input sample, T_i being the *ith* decision tree and B is the total number of trees in RF. The output for feature i can be evaluated from Eq. (20) by averaging the votes from individual decision trees.

$$Output_i = \frac{1}{B} \times \sum_{i=1}^{B} T_i(x'). \tag{20}$$

For the *ith* tree, a random vector $\mathbf{r}_i$ is generated, which has the same distribution but is independent of all past random vectors $\mathbf{r}_1, \ldots, \mathbf{r}_{i-1}$ and afterward a tree T_i is grown by utilizing $\mathbf{r}_i$ and the training sequence, which results in a classifier $h(x, \mathbf{r}_i)$. An RF can be considered as a classifier including several tree-structured classifiers, i.e., $\{h(x, \mathbf{r}_i), i = 1, \ldots, B\}$. Binary decisions are made if the desired conditions are met, i.e., $x_i < Threshold$ or vice versa.

A margin function can be defined, which can measure the degree of correctly classified outcomes that exceed the average votes related to any other class in the dependent variable. Consider $\mathbf{X}$ to be a random vector sampled from training data, Y being the classification response and θ_k to be the parameters of the decision tree containing the tree structure for classifier $h_k(x)$.

$$\theta_k = \theta_{k1}, \theta_{k2}, \ldots, \theta_{kB}, \tag{21}$$

$$mg(\mathbf{X}, Y) = \frac{\sum_{i=1}^{B} I(h_k(\mathbf{X}) = Y)}{B} - \max_{j \neq y} \left[\frac{\sum_{i=1}^{B} I(h_k(\mathbf{X}) = j)}{B} \right]. \tag{22}$$

From Eq. (22), $I(.)$ is the Indicator function.

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

The margin function shares outputs according to the specific conditions (i.e., in case the set of classifiers make a correct classification $mg\,(\mathbf{X}, Y) > 0$). However for an incorrect classification $mg\,(\mathbf{X}, Y) < 0$, the higher value of margin function indicates a greater degree of confidence in the classification.

The generalization error or the misclassification rate PE^*, over the space $\mathbf{X}$, Y can be given by

$$PE^* = P_{\mathbf{X},Y}(mg(\mathbf{X}, Y) < 0). \tag{24}$$

For a large Random Forest, i.e., $B \to \infty$, the generalization error can be evaluated from Eq. (25), which has a limiting value due to the constraint that the RF cannot over fit data.

$$PE^* \to P_{\mathbf{X},Y}(P_\theta(h(X,\theta) = Y)) - \max_{j \neq y} P_\theta(h(X,\theta) = j) < 0. \tag{25}$$

The strength s of classifiers (h_k) is expected to be an estimation of the accuracy of individual trees in the forest.

$$s = E_{\mathbf{X},Y}(mr(\mathbf{X}, Y)). \tag{26}$$

4. HANDWRITTEN EXPERIMENTAL DATA

Handwritten numerals vary due to different reasons some of which were mentioned previously. The novel method proposed in this paper was validated and tested on well-known databases. When a database was not readily available, such as the case for Urdu, a database was specifically developed for that language and named PMU-UD. The first database used in this study for Eastern Arabic digits is the Modified Arabic Handwritten Digits Database (MADBase). MADBase was developed by collecting samples from 700 participants. It contains 70,000 numerals with resolution of 300 dpi at 28 × 28 pixels [29]. The second database used for Western Arabic (English) was the Modified National Institute of Standards and Technology Database (MNIST). MNIST contains 70,000 patterns collected from 250 writers [30]. For Persian, the well-known HODA database was used containing 80,000 different numeral patterns [31]. ICDAR was used for Bengali numerals. It is a small dataset containing 1,700 numeral patterns [32]. For Devanagari, the Devanagari Handwritten Character Dataset (DHCD) was used which consists of 20,000 numeral patterns [33]. Since the authors were unable to obtain a dataset for Urdu, a dataset was developed. The dataset was collected from 170 participants with a total of 5,180 numeral patterns. The dataset is named Prince Mohammad Bin Fahd University - Urdu Database (PMU-UD) [34]. The participants were asked to write the numerals from 0–9 five times each. Participants age ranged from 25 to 55 years old. Table II details the different datasets used in this study to validate and test the novel method.

5. RESULTS AND DISCUSSIONS

Through a carefully designed process that included the three phases listed above, the proposed method was tested on the well-known datasets listed in Table II above. Digits went through the preprocessing phase listed above and then the novel feature extraction method was applied to extract 65 different features of the numerals that are detailed in a previous section. Four different prominent classifiers were used to test the accuracy of the numeral recognition process: AIRS, MLP, Logistic and RF. As shown in Table III, RF

Table II. Summary of handwritten experimental datasets of different languages.

Numeral Language	Database	Data Source	Training Dataset	Testing Dataset	Total Patterns
Eastern Arabic	MADBase	700 Participants	60,000	10,000	70,000
Western Arabic (English)	MNIST	250 Writers	60,000	10,000	70,000
Persian	HODA	12,000 Registration Forms	60,000	20,000	80,000
Urdu	PMU-UD	170 Participants	3,500	1,680	51,80
Bengali	ICDAR	35 Writers	1,500	200	1,700
Devanagari	DHCD	NA	17,000	3,000	20,000

Table III. Recognition accuracy rates for different databases with four different classifiers.

	Artificial Immune Recognition System	Multi-Layer Perceptron	Logistic	Random Forest
Eastern Arabic	96.25%	97.30%	96.87%	98.10%
Western Arabic (English)	90.00%	94.40%	91.36%	95.31%
Persian	92.91%	96.66%	95.41%	96.92%
Urdu	93.72%	95.20%	93.10%	97.27%
Bengali	90.62%	92.27%	94.16%	95.98%
Devanagari	91.80%	96.66%	92.90%	96.80%
Average	92.55%	95.42%	93.97%	96.73%

Table IV. Recognition accuracy rates for different languages.

Source	Method	Dataset Language	Accuracy
Alkhateeb and Alseid [7]	DBN Classifier with DCT Features	Arabic data of 70,000 records written by 700 (ADBase)	85.26%
Salimi and Giveki [8]	SVD and PSO based Classifier	Persian/Farsi Dataset (HODA)	97.02%
Musleh et al. [9]	Fuzzy Logic and SVM	Arabic Dataset of 32,695 digits	98.01%
Hosseinzadeh et al. [10]	Ensemble Projection Feature	English MNIST, USPS, and ICDAR datasets	89.86%
Sarkhel et al. [14]	Sorting Harmony-Search Algorithm (NSHA)	Bangali Numerals Dataset	98.00%
Basu et al. [15]	Multi-Layer Perceptron (MLP)	6000 handwritten Bangla digits	95.10%
Wang et al. [16]	DCT and HMM	English MNIST dataset	97.20%

achieved the highest accuracy for all datasets in all languages. Table IV summarizes the recognition accuracy rates for different languages proposed in recent studies which are also discussed in the literature review section. For MADBase, RF achieved an accuracy of 98.1%, whereas a 95.31% accuracy was achieved for MNIST using RF. Moreover, a 96.92% accuracy was obtained for HODA and a 97.27% accuracy was found for PMU-UD using RF. Similarly, an accuracy of 95.98% and 96.8% was achieved for ICDAR and DHCD, respectively, using RF. Overall it can be concluded that RF achieved the best recognition rate for all datasets in all languages using the novel proposed feature extraction method detailed in this paper. The RF achieved an average accuracy rate of 96.73% for the multi-language feature extraction method proposed.

Figure 4 illustrates the cumulative error rate comparison of target and the predicted digits. The average error rate for each digit in each language was calculated for all classifiers. RF produces overall low error rate for every digit in every language targeted in this study. For example, in RF, the targeted digit 0 was mistakenly recognized 0.72% of the times as the number 5, 0.09% of times as the number 4, and 0.07% of the times as the number 3. In the same classifier RF, the target number 2 was mistakenly recognized 1.45% of times as the digit 3. Fig. 4 shows that the confusion between digits is at its lowest using the RF Classifier. This confusion is at its highest using the AIRS classifier. Overall, the RF classifier is shown to produce the best accuracy by producing the lowest error rates. At the level of individual languages, some of the above results are below the accuracy rates found in the literature. However, the goal of this work is to achieve the highest recognition accuracy for multi-language numerals.

Figure 5 shows the average accuracy of each digit in all languages and classifiers used in the study. Even though on an individual digit-by-digit comparison, some classifiers produced better results than RF, RF generated the highest

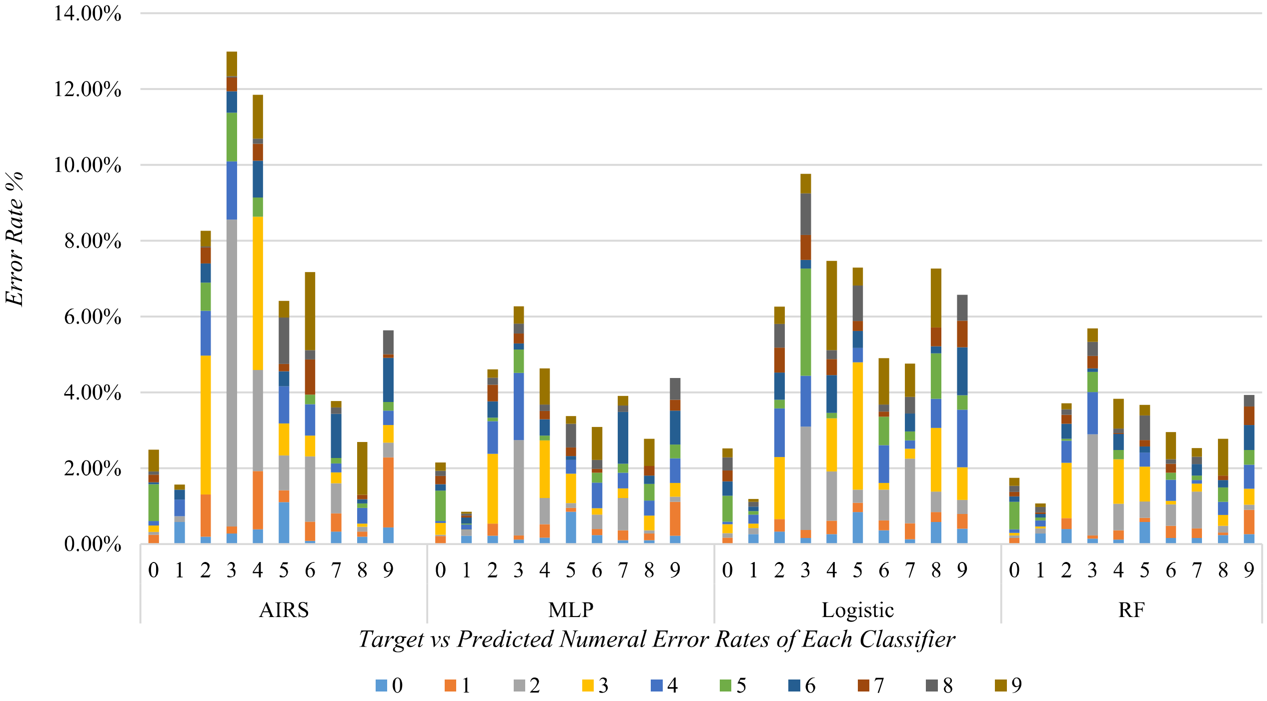
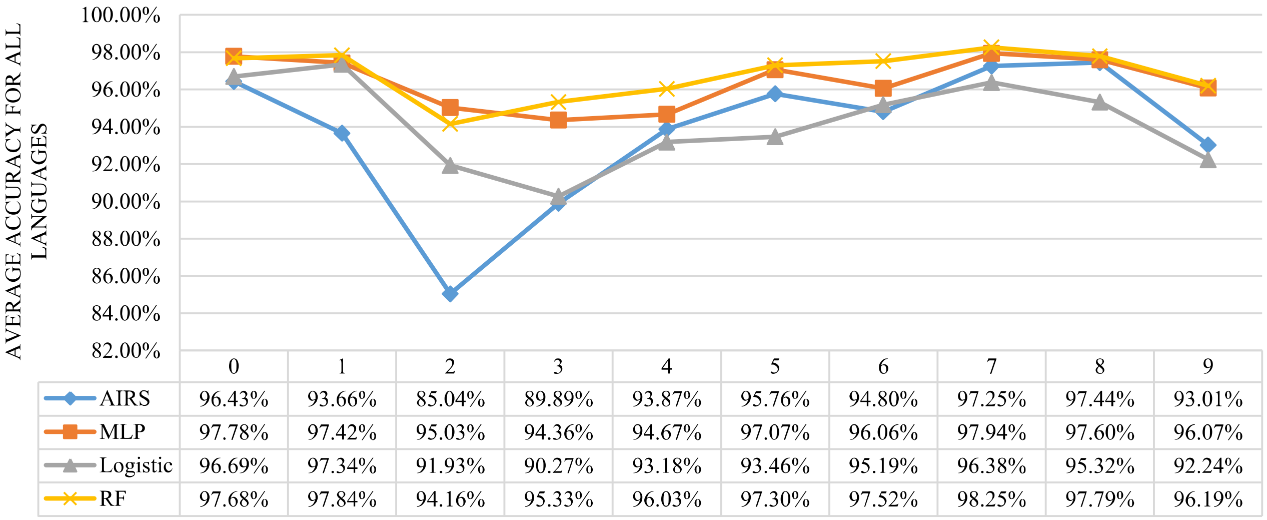


Figure 4. Comparison of target vs predicted digit cumulative error rate of the selected classifiers.



	0	1	2	3	4	5	6	7	8	9
AIRS	96.43%	93.66%	85.04%	89.89%	93.87%	95.76%	94.80%	97.25%	97.44%	93.01%
MLP	97.78%	97.42%	95.03%	94.36%	94.67%	97.07%	96.06%	97.94%	97.60%	96.07%
Logistic	96.69%	97.34%	91.93%	90.27%	93.18%	93.46%	95.19%	96.38%	95.32%	92.24%
RF	97.68%	97.84%	94.16%	95.33%	96.03%	97.30%	97.52%	98.25%	97.79%	96.19%

Figure 5. Comparison of average accuracy for all languages.

accuracy for all the digits combined. For example, for digit 0, MLP produced the highest recognition accuracy of 97.78% exceeding that of RF by 0.1%. This pattern was repeated for digit 2. However, for all other digits 1,3,4,5,6,7,8, and 9, RF exceeded all other classifiers producing a recognition accuracy rate of 97.84%, 95.33%, 96.03%, 97.3%, 97.52%, 98.25%, 97.79% and 96.19%, respectively. Fig. 2 illustrates that the RF generates superior results as compared to all other classifiers used in this study.

Due to the resemblance of the numerals in Arabic, Persian, and Urdu with only a few digits that are different, we combined the dataset MADBase, HODA and PMU-UD into one mega database, and applied the same three-phase process of preprocessing, feature extraction, and classification. The combined dataset was tested using the proposed feature extraction method and the four classifiers. RF again produced the best recognition accuracy (97.26%) for the combined dataset compared with 92.93% for AIRS, 95.88% for MLP, and 92.8% for Logistic.

No explicit comparison with previous methods of numeral recognition was made since the proposed unified recognition system for multi-language numerals is a novelty.

If a comparison is made between the proposed method and those applicable to one language only, it is obvious that a machine learning system for recognizing numerals in one language can achieve higher recognition rate.

6. CONCLUSION

In this paper, we proposed a novel Local Feature Extraction method that is used to design a unified multi-language handwritten numeral recognition system. We targeted many languages even though their digits do not resemble each other. In this study, we proposed 65 geometrically based local features. Additionally, in this paper, we also proposed and showed that the RF classifier in conjunction with the proposed Local Feature Extraction method yield optimum results. The proposed method is tested on six different well-known databases of different languages by using RF. An average recognition rate of 96.73% was achieved for the recognition of handwritten numerals of six different languages. These rates exceed other methods' rates reported in the extant literature and establish a good launchpad for future work in the development of a unified system for the recognition of handwritten numerals in other languages. It can also be observed that the proposed method produced very low error rates and very low confusion rates with other digits. Future work may include using fuzzy logic to further reduce the confusion between different digits, thereby increasing the recognition accuracy even further. Future work will evaluate modification of this proposed system to detect digits in other languages, including Gujarati, Gurmucki, Kannada, Lao, Limbo, Malayalam, Mongolian, Myanmar, Oriya, Telugu, Thai, and Tibetan. The possibility of redesigning the system in a cloud-based environment will also be part of future work in order to achieve a continuous learning curve and obtain a continuous accuracy improvement.

ACKNOWLEDGMENTS

This work is supported by Prince Mohammad Bin Fahd University (PMU) Phase 2 Research Grant. The authors would like to thank Prince Mohammad Bin Fahd University for supporting this research project.

REFERENCES

[1] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 63–84 (2000).

[2] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," *Advances in Neural Information Processing Systems* (2009), pp. 545–552. Q.5

[3] M. S. S. Mohamed, B. M. T. Shamsul, R. Rahman, M. S. Aini, and N. A. A. Jalil, "Integrating Usability in Automotive Navigation User Interface Design via Kansei Engineering," Modern Appl. Sci. **10**, 208 (2016).

[4] A. Riener, A. Ferscha, F. Bachmair, P. Hagmüller, A. Lemme, D. Muttenthaler, and F. Weger, "Standardization of the in-car gesture interaction space," *Proc. of the 5th Int'l. Conf. on Automotive User Interfaces and Interactive Vehicular Applications* (ACM, 2013), pp. 14–21.

[5] S. Kiljan, K. Simoens, D. D. Cock, M. V. Eekelen, and H. Vranken, "A survey of authentication and communications security in online banking," ACM Comput. Surv. **49**, 61 (2016).

[6] P. Saravanan, S. Clarke, D. H. P. Chau, and H. Zha, "Latent gesture: Active user authentication through background touch analysis," *Proc. of the Second Int'l. Symposium of Chinese CHI* (ACM, 2014), pp. 110–113.

[7] J. H. AlKhateeb and M. Alseid, "DBN-Based learning for Arabic handwritten digit recognition using DCT features," *6th Int'l. Conf. on Computer Science and Information Technology (CSIT)* (IEEE, Piscataway, NJ, 2014), pp. 222–226.

[8] H. Salimi and D. Giveki, "Farsi/Arabic handwritten digit recognition based on ensemble of SVD classifiers and reliable multi-phase PSO combination rule," Int. J. Doc. Anal. Recognit. **16**, 371–386 (2013).

[9] D. Musleh, K. Halawani, and S. Mahmoud, *Fuzzy Modeling for Handwritten Arabic Numeral Recognition*, vol. 14, pp. 1–10. Q.6

[10] S. A. Azeem, M. El Meseery, and H. Ahmed, "Online arabic handwritten digits recognition," *Int'l. Conf. on Frontiers in Handwriting Recognition (ICFHR)* (IEEE, Piscataway, NJ, 2012), pp. 135–140.

[11] J. Sadri, M. R. Yeganehzad, and J. Saghi, "A novel comprehensive database for offline Persian handwriting recognition," Pattern Recognit. **60**, 378–393 (2016).

[12] A. Boukharouba and A. Bennia, "Novel feature extraction technique for the recognition of handwritten digits," Appl. Comput. Inform. **13**, 19–26 (2015).

[13] H. Karimi, A. Esfahanimehr, M. Mosleh, S. Salehpour, and O. Medhati, "Persian handwritten digit recognition using ensemble classifiers," Procedia Computer Science **73**, 416–425 (2015).

[14] R. Sarkhel, N. Das, A. K. Saha, and M. Nasipuri, "A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition," Pattern Recognit. **58**, 172–189 (2016).

[15] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, and D. Basu, "Handwritten Bangla digit recognition using classifier combination through DS technique," Pattern Recognit. Mach. Intell. 236–241 (2005).

[16] W. Wang, Y. Li, M. Wang, L. Wang, Q. Liu, W. Banerjee, and M. Liu, "A hardware neural network for handwritten digits recognition using binary RRAM as synaptic weight element," *Silicon Nanoelectronics Workshop (SNW)* (IEEE, Piscataway, NJ, 2016), pp. 50–51.

[17] S. S. Ali and M. U. Ghani, "Handwritten digit recognition using DCT and HMMs," *12th Int'l. Conf. Frontiers of Information Technology (FIT)* (IEEE, Piscataway, NJ, 2014), pp. 303–306.

[18] M. Jie, I. A. Aziz, H. Hasbullah, and S. A. B. Azizan, "Handwritten digits recognition based on improved label propagation algorithm," *3rd Int'l. Conf. on Computer and Information Sciences (ICCOINS)* (IEEE, Piscataway, NJ, 2016), pp. 345–350.

[19] R. Bajaj, L. Dey, and S. D. Chaudhury, "numeral recognition by combining decision of multiple connectionist classifiers," Sadhana **27**, 59–72 (2002).

[20] M. O. Al_barraq and S. C. Mehrotra, "Recognition of Arabic Handwritten Amount in Cheque through Windowing Approach.," Int. J. Comput. Appl. **115** (2015).

[21] J. L. Fan and F. Zhao, "Two-dimensional Otsu's curve thresholding segmentation method for gray-Level images," Dianzi Xuebao (Acta Electronica Sinica) **35**, 751–755 (2007).

[22] G. G. Rajput and H. B. Anita, "Handwritten script recognition using DCT and wavelet features at block level," IJCA, Special issue on RTIPPR **3**, 158–163 (2010).

[23] J. Cao, M. Ahmadi, and M. Shridhar, "Recognition of handwritten numerals with multiple feature and multistage classifier," Pattern Recognit. **28**, 153–160 (1995).

[24] U. Aickelin, D. Dasgupta, and F. Gu, "Artificial immune systems," *Search Methodologies* (Springer, US), pp. 187–211. Q.7

[25] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," Atmos. Environ. **32**, 2627–2636 (2014).

[26] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression," Geosci. Remote Sens. Lett. **10**, 318–322 (2013).

[27] A. Liaw and M. Wiener, “Classification and regression by randomForest,” R news **2**, 18–22 (2002).

[28] C. Strobl, J. Malley, and G. Tutz, “An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests,” Psychological Methods **14**, 323 (2009).

[29] S. Abdelazeem, “Comparing Arabic and Latin handwritten digits recognition problems,” *World Academy of Science, Engineering and Technology*, 54.

[30] L. Deng, “The MNIST database of handwritten digit images for machine learning research [best of the web],” IEEE Signal Process. Mag. **29**, 141–142 (2009).

[31] H. Khosravi and E. Kabir, “Introducing a very large dataset of handwritten Farsi digits and a study on their varieties,” Pattern Recognit. Lett. **28**, 1133–1141 (2007).

[32] P. Purkait and B. Chanda, “Off-line recognition of hand-written Bengali numerals using morphological features,” *Int’l. Conf. on Frontiers in Handwriting Recognition (ICFHR)* (IEEE, Piscataway, NJ, 2010), pp. 363–368.

[33] S. Acharya, A. K. Pant, and P. K. Gyawali, “Deep learning based large scale handwritten Devanagari character recognition,” *9th Int’l. Conf. on Software, Knowledge, Information Management and Applications (SKIMA)* (IEEE, Piscataway, NJ, 2015), pp. 1–6.

[34] PMU-UD database Repository Web URL: https://sourceforge.net/projects/pmu-ud/.