

## Research Article

# A Study of Moment Based Features on Handwritten Digit Recognition

**Pawan Kumar Singh, Ram Sarkar, and Mita Nasipuri**

*Department of Computer Science and Engineering, Jadavpur University, 188 Raja S. C. Mullick Road, Kolkata, West Bengal 700032, India*

Correspondence should be addressed to Pawan Kumar Singh; pawansingh.ju@gmail.com

Received 3 November 2015; Revised 16 January 2016; Accepted 27 January 2016

Academic Editor: Miin-Shen Yang

Copyright © 2016 Pawan Kumar Singh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Handwritten digit recognition plays a significant role in many user authentication applications in the modern world. As the handwritten digits are not of the same size, thickness, style, and orientation, therefore, these challenges are to be faced to resolve this problem. A lot of work has been done for various non-*Indic* scripts particularly, in case of *Roman*, but, in case of *Indic* scripts, the research is limited. This paper presents a script invariant handwritten digit recognition system for identifying digits written in five popular scripts of Indian subcontinent, namely, *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu*. A 130-element feature set which is basically a combination of six different types of moments, namely, geometric moment, moment invariant, affine moment invariant, Legendre moment, Zernike moment, and complex moment, has been estimated for each digit sample. Finally, the technique is evaluated on *CMATER* and *MNIST* databases using multiple classifiers and, after performing statistical significance tests, it is observed that Multilayer Perceptron (MLP) classifier outperforms the others. Satisfactory recognition accuracies are attained for all the five mentioned scripts.

## 1. Introduction

The field of automated reading of printed or handwritten documents by the electronic devices is known as Optical Character Recognition (OCR) system, which is broadly defined as the process of recognizing either printed or handwritten text from document images and converting it into electronic form. OCR systems can contribute tremendously to the advancement of the automation process and can improve the interaction between man and machine in many applications, including office automation, bank check verification, postal automation, and a large variety of business and data entry applications. Handwritten digit recognition is the method of recognizing and classifying handwritten digits from 0 to 9 without human interaction [1]. Although the recognition of handwritten numerals has been studied for more than three decades and many techniques with high accuracy rates have already been developed, the research in this area continues with the aim of improving the recognition rates further.

Handwritten digit recognition is a complex problem due to the fact that variation exists in writing style of different

writers. The phenomenon that makes the problem more challenging is the inherent variation in writing styles at different instances. Due to this reason, building a generic recognizer that is capable of recognizing handwritten digits written by diverse writers is not always feasible [2]. However, the extraction of the most informative features with highly discriminatory ability to improve the classification accuracy with reduced complexity remains one of the most important problems for this task. It is a task of great importance for which there are standard databases that allow different approaches to be compared and validated.

India is a multilingual country with 23 constitutionally recognized languages written in 12 major scripts [1]. Besides these, hundreds of other languages are used in India, each one with a number of dialects. The officially recognized languages are *Hindi*, *Bengali*, *Punjabi*, *Marathi*, *Gujarati*, *Oriya*, *Sindhi*, *Assamese*, *Nepali*, *Urdu*, *Sanskrit*, *Tamil*, *Telugu*, *Kannada*, *Malayalam*, *Kashmiri*, *Manipuri*, *Konkani*, *Maithili*, *Santhali*, *Bodo*, *English*, and *Dogri*. The 12 major scripts used to write these languages are *Devanagari*, *Bangla*, *Oriya*, *Gujarati*, *Gurumukhi*, *Tamil*, *Telugu*, *Kannada*, *Malayalam*, *Manipuri*,

*Roman*, and *Urdu*. In a multilingual country like India, it is a common scenario that a document like job application form, railway ticket reservation form, and so forth is composed of text contents written in different languages/scripts in order to reach a larger cross section of people. The variation of different scripts may be in the form of numerals or alpha numerals in a single document page. But the techniques developed for text identification generally do not incorporate the recognition of digits. This is because the features required for the text identification may not be applicable for identifying the digits.

The paper is organized as follows: Section 2 presents a brief review of some of the previous approaches to handwritten digit recognition whereas, in Section 3, we introduce our script independent handwritten digit recognition system. Section 4 describes the performance of our system on realistic databases of handwritten digits and, finally, Section 5 concludes the paper.

## 2. Review of Related Works

Gorgevik and Cakmakov [3] developed Support Vector Machine (SVM) based digits recognition system for handwritten *Roman* numerals. They extracted four types of features from each digit image: (1) projection histograms, (2) contour profiles, (3) ring-zones, and (4) Kirsch features. They reported 97.27% recognition accuracy on National Institute of Standards and Technology (NIST) handwritten digits database [4]. In [5], Chen et al. proposed max-min posterior pseudoprobabilities framework for *Roman* handwritten digit recognition. They extracted 256 dimension directional features from the input image. Finally, these features were transformed into a set of 128 features using Principal Component Analysis (PCA). They reported recognition accuracy of 98.76% on NIST database [4]. Labusch et al. [6] described a sparse coding based feature extraction method with SVM as a classifier. They found recognition accuracy of 99.41% on MNIST (Modified NIST) handwritten digits database [7]. The work described in [8] combined three recognizers by majority vote, and one of them is based on Kirsch gradient (four orientations), dimensionality reduction by PCA, and classification by SVM. They achieved an accuracy rate of 95.05% with 0.93% error on 10,000 test samples of MNIST database [7]. Mane and Ragha [9] performed handwritten digit recognition using elastic image matching technique based on eigendeformation, which is estimated by the PCA of actual deformations automatically selected by the elastic matching. They achieved an overall accuracy of 94.91% on their own database collected from different individuals of various professions for the experiment. Cruz et al. [10] presented a handwritten digit recognition system which uses multiple feature extraction methods and classifier ensemble. A total of six feature extraction algorithms, namely, Multizoning, Modified Edge Maps, Structural Characteristics, Projections, Concavities Measurements, and Gradient Directional, were evaluated in this paper. A scheme using neural networks as a combiner achieved a recognition rate of 99.68% on a training set of 60,000 images and a test set of 10,000 images of MNIST database.

Dhendra et al. [11] investigated a script independent automatic numeral recognition system for recognition of *Kannada*, *Telugu*, and *Devanagari* handwritten numerals. In the proposed method, 30 classes were reduced to 18 classes by extracting the global and local structural features like directional density estimation, water reservoirs, maximum profile distances, and fill-hole density. Finally, a probabilistic neural network (PNN) classifier was used for the recognition system which yielded an accuracy of 97.20% on a total of 2550 numeral images written in *Kannada*, *Telugu*, and *Devanagari* scripts. In [12], Yang et al. proposed supervised matrix factorization method used directly as multiclass classifier. They reported recognition accuracy of 98.71% with supervised learning approach on MNIST database [7]. In [13], a mixture of multiclass logistic regression models was described. They claimed recognition accuracy of 98% on the Indian digit database provided by CENPARMI [14]. Das et al. [15] described a technique for creating a pool of local regions and selection of an optimal set of local regions from that pool for extracting optimal discriminating information for handwritten *Bangla* digit recognition. Genetic algorithm (GA) was then applied on these local regions to sample the best discriminating features. The features extracted from these selected local regions were then classified with SVM and recognition accuracy of 97% was achieved. In [16], a wavelet analysis based technique for feature extraction was reported. For classification, SVM and  $k$ -Nearest Neighbor ( $k$ -NN) were used and an overall recognition accuracy of 97.04% was reported on MNIST digit database [7]. A comparative study in [17] was conducted by training the neural network using Backpropagation (BP) algorithm and further using PCA for feature extraction. Digit recognition was finally carried out using 13 algorithms, neural network algorithm, and the Fisher Discriminant Analysis (FDA) algorithm. The FDA algorithm proved less efficient with an overall accuracy of 77.67%, whereas the BP algorithm with PCA for its feature extraction gave an accuracy of 91.2%.

In [18], a set of structural features (namely, number of holes, water reservoirs in four directions, maximum profile distances in four directions, and fill-hole density) and  $k$ -NN classifier were employed for classification and recognition of handwritten digits. They reported recognition accuracy of 96.94% on 5000 samples of MNIST digit database [7]. In [19], AlKhateeb and Alseid proposed an Arabic handwritten digit recognition system using Dynamic Bayesian Network. They employed DCT coefficients based features for classification. The system was tested on *Indo-Arabic* digits database (ADBase) which contains 70,000 *Indo-Arabic* digits [20] and an average recognition accuracy of 85.26% was achieved on 10,000 samples. Ebrahimzadeh and Jampour [21] proposed an appearance feature-based approach using Histogram of Oriented Gradients (HOG) for handwritten digit recognition. A linear SVM was then used for classification of the digits in MNIST dataset and an overall accuracy of 97.25% had been realized. Gil et al. [22] presented a novel approach using SVM binary classifiers and unbalanced decision trees. Two classifiers were proposed in this study where one used the digit characteristics as input and the other used the whole image as such. It is observed that a handwritten

digit recognition accuracy of 100% was achieved on MNIST database using the whole image as input. El Qacimy et al. [23] investigated the effectiveness of four feature extraction approaches based on Discrete Cosine Transform (DCT), namely, DCT upper left corner (ULC) coefficients, DCT zigzag coefficients, block based DCT ULC coefficients, and block based DCT zigzag coefficients. The coefficients of each DCT variant were used as input data for SVM classifier and it was found that block based DCT zigzag feature extraction yielded a superior recognition accuracy of 98.76% on MNIST database. AL-Mansoori [24] implemented a MLP classifier to recognize and predict handwritten digits. A dataset of 5000 samples were obtained from MNIST database and an overall accuracy of 99.32% was achieved.

From the above literature, it is clear that most of the works have been done for the *Roman* script, whereas relatively few works [11, 15, 19] have been reported for the digit recognition written in *Indic* scripts. The main reasons for this slow progress could be attributed to the complexity of the shape of *Indic* scripts as opposed to *Roman* script. Again, the discriminating power of the features exploited till now is not easily measurable; investigative experimentations will be necessary for identifying new feature descriptors for effective classification of complex handwritten digits of different scripts. It is also revealed that the methods, described in the literature, suffer from larger computational time mainly due to feature extraction from large dataset. In addition, the above recognition systems fail to meet the desired accuracy when exposed to different multiscript scenario. Hence, it would be beneficial for multilingual country like India if there is a method which is independent of script and yields reasonable recognition accuracy. This has motivated us to introduce a script invariant handwritten digit recognition system for identifying digits written in five popular scripts, namely, *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu*. The key module of the proposed methodology is shown in Figure 1.

### 3. Feature Extraction Methodology

One of the basic problems in the design of any pattern recognition system is the selection of a set of appropriate features to be extracted from the object of interest. Research on the utilization of moments for object characterization in both invariant and noninvariant tasks has received considerable attention in recent years. Describing digit images with moments instead of other more commonly used pattern recognition features (described in [21–23]) means that global properties of the digit image are used rather than local properties. So, for the present work, we considered a moment based approach which is described in the next subsection.

**3.1. Moments.** Moments are pure statistical measure of pixel distribution around the center of gravity of the image and allow capturing global shapes information [25]. They describe numerical quantities at some distance from a reference point or axis. Moments are commonly used in statistics to characterize the distribution of random variables and,

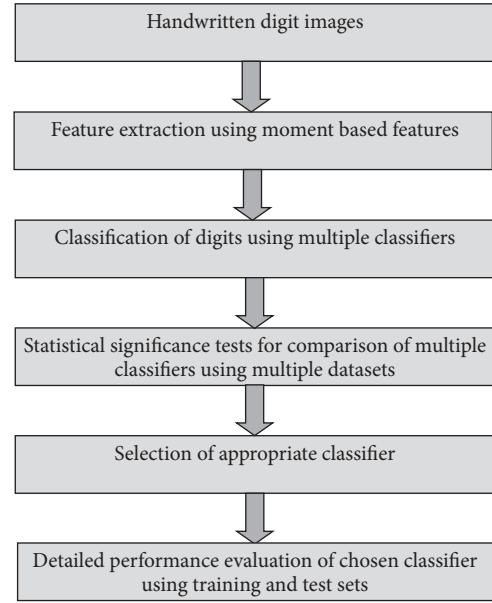


FIGURE 1: Schematic diagram illustrating the key modules of the proposed methodology.

similarly, in mechanics to characterize bodies by their spatial distribution of mass.

A complete characterization of moment functional over a class of univariate functions was given by Hausdorff [26] in 1921.

Let  $\{\mu_n\}$  be a real sequence of numbers and let us define

$$\Delta^m \mu_n = \sum_{i=0}^m (-1)^i \binom{m}{i} \mu_{n+i}. \quad (1)$$

Note that  $\Delta^m \mu_n$  can be viewed as the  $m$ th order derivative of  $\mu_n$ .

By the Hausdorff theorem, a necessary and sufficient condition that there exists a monotonic function  $F(x)$  satisfying the system

$$\mu_n = \int_0^1 x^n dF(x), \quad n = 0, 1, 2, \dots \quad (2)$$

is that the system of linear inequalities

$$\Delta^k \mu_n \geq 0 \quad k = 0, 1, 2, \dots \quad (3)$$

should be satisfied; that is, if  $f(x)$  is a positive function (in case of image processing), then the set of functionals

$$\int_0^1 x^n f(x) dx, \quad n = 0, 1, \dots \quad (4)$$

completely characterizes the function.

A necessary and sufficient condition that there exists a function  $F(x)$  of bounded variation satisfying (7) is that the sequence

$$\sum_{m=0}^p \binom{p}{m} |\Delta^{p-m} \mu_m| \quad p = 0, 1, 2, \dots \quad (5)$$

should be bounded. The use of moments for image analysis is straightforward if we consider a binary or gray level image segment as a two-dimensional density distribution function. It can be assumed that an image can be represented by a real-valued measurable function  $f(x, y)$ . In this way, moments may be used to characterize an image segment and extract properties that have analogies in statistics and mechanics. In image processing and computer vision, an image moment is a certain particular weighted average (moment) of the image pixels' intensities or a function of such moments, usually chosen to have some attractive property or interpretation. The first significant work considering moments for pattern recognition was performed by Hu [27]. He derived relative and absolute combinations of moment values that are invariant with respect to scale, position, and orientation based on the theories of invariant algebra that deal with the properties of certain classes of algebraic expressions which remain invariant under general linear transformations. Size invariant moments are derived from algebraic invariants but can be shown to be the result of simple size normalization. Translation invariance is achieved by computing moments that have been translated by the negative distance to the centroid and thus normalized so that the center of mass of the distribution is at the origin (central moments).

**3.2. Geometric Moments.** Geometric moments are defined as the projection of the image intensity function  $f(x, y)$  onto the monomial  $x^p y^q$  [25]. The  $(p + q)$ th order geometric moment  $M_{pq}$  of a gray level image  $f(x, y)$  is defined as

$$M_{pq} = \iint_{-\infty}^{\infty} x^p y^q f(x, y) dx dy, \quad (6)$$

where  $p, q = 0, 1, 2, \dots, \infty$ . Note that the monomial product  $x^p y^q$  is the basis function for this moment definition. A set of  $n$  moments consists of all  $M_{pq}$ 's for  $p + q \leq n$ ; that is, the set contains  $(1/2)(n + 1)(n + 2)$  elements. If  $f(x, y)$  is piecewise continuous and contains nonzero values only in a finite region of the  $xy$ -plane, then the moment sequence  $\{M_{pq}\}$  is uniquely determined by  $f(x, y)$  and, conversely,  $f(x, y)$  is uniquely determined by  $\{M_{pq}\}$ . Considering the fact that an image segment has finite area or in the worst case is piecewise continuous, moments of all orders exist and a complete moment set can be computed and used uniquely to describe the information contained in the image. However, obtaining all the information contained in the image requires an infinite number of moment values. Therefore, to select a meaningful subset of the moment values that contain sufficient information to characterize the image uniquely for a specific application becomes very important. In case of a digital image of size  $M \times N$ , the double integral in (6) is replaced by a summation which turns into this simplified form:

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p y^q f(x, y), \quad (7)$$

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

When  $f(x, y)$  changes by translating, rotating, or scaling, then the image may be positioned such that its *center of mass*

(COM) is coincided with the origin of the field of view, that is,  $(\bar{x} = 0)$  and  $(\bar{y} = 0)$  and then the moments computed for that object are referred to as *central moment* [25] and it is designated by  $\mu_{pq}$ . The simplified form of central moment of order  $(p + q)$  is defined as follows:

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^p (y - \bar{y})^q f(x, y), \quad (8)$$

where  $\bar{x} = m_{10}/m_{00}$  and  $\bar{y} = m_{01}/m_{00}$ .

The pixel point  $(\bar{x}, \bar{y})$  is the COM of the image. The *central moments*  $\mu_{pq}$  computed using the centroid of the image are equivalent to  $m_{pq}$  whose center has been shifted to centroid of the image. Therefore, the *central moments* are invariant to image translations. Scale invariance can be obtained by normalization. The normalized *central moments* denoted by  $\eta_{pq}$  are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad (9)$$

where  $\gamma = (p + q)/2 + 1$  for  $(p + q) = 2, 3, \dots$

The second order moments,  $\{\eta_{02}, \eta_{11}, \eta_{20}\}$  known as the *moments of inertia*, may be used to determine an important image feature called orientation [25]. Here, the feature values  $F1$ – $F3$  have been computed from *moments of inertia* of the word images. In general, the orientation of an image describes how the image lies in the field of view or the directions of the principal axes. In terms of moments, the orientation of the principal axis,  $\theta$ , taken as feature value  $F4$ , is given by

$$\theta = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right), \quad (10)$$

where  $\theta$  is the angle of the principal axis nearest to the  $x$ -axis and is in the range  $-\pi/4 \leq \theta \leq \pi/4$ . The minimum and maximum distances ( $r_{\min}$  and  $r_{\max}$ ) between the centroid and the boundary of an image are also feature descriptors. The ratio  $r_{\max}/r_{\min}$  is called *elongation* or *eccentricity* ( $F5$ ) and can be defined in terms of *central moments* as follows:

$$e = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{\mu_{00}}. \quad (11)$$

**3.3. Moment Invariants.** Based on the theory of algebraic invariants, Hu [27] derived relative and absolute combinations of moments that are invariant with respect to scale, position, and orientation. The method of moment invariants is derived from algebraic invariants applied to the moment generating function under a rotation transformation. The set of absolute moment invariants consists of a set of nonlinear combinations of central moment values that remain invariant under rotation. A set of seven *invariant moments* can be derived based on the normalized *central moments* of order

three that are invariant with respect to image scale, translation, and rotation. Consider

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

This set of moments is invariant to translation, scale change, mirroring (within a minus sign), and rotation. The 2D moment invariant gives seven features (F6–F12) which had been used for the current work.

**3.4. Affine Moment Invariants.** The affine moment invariants are derived to be invariants to translation, rotation, and scaling of shapes and under 2D Affine transformation. The six affine moment invariants [28] used for the present work are defined as follows:

$$\begin{aligned}
I_1 &= \frac{1}{\mu_{00}^4} (\mu_{20}\mu_{02} - \mu_{11}^2), \\
I_2 &= \frac{1}{\mu_{00}^{10}} (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 \\
&\quad + 4\mu_{03}\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2), \\
I_3 &= \frac{1}{\mu_{00}^7} (\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) \\
&\quad + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)), \\
I_4 &= \frac{1}{\mu_{00}^{11}} (\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{21}\mu_{02}\mu_{03} \\
&\quad + 9\mu_{20}^2\mu_{02}\mu_{12}^2 + 12\mu_{20}\mu_{02}^2\mu_{11}\mu_{03}\mu_{21} \\
&\quad + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} \\
&\quad - 8\mu_{11}^3\mu_{03}\mu_{30} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + 9\mu_{20}\mu_{02}^2\mu_{21}^2)
\end{aligned}$$

$$+ 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - 6\mu_{11}^2\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^2\mu_{30}^2),$$

$$I_5 = \frac{1}{\mu_{00}^6} (\mu_{40}\mu_{04} - 4\mu_{31}\mu_{13} + 3\mu_{22}^2),$$

$$\begin{aligned}
I_6 &= \frac{1}{\mu_{00}^9} (\mu_{40}\mu_{04}\mu_{22} + 2\mu_{31}\mu_{22}\mu_{13} - \mu_{40}\mu_{13}^2 - \mu_{04}\mu_{31}^2 \\
&\quad - \mu_{22}^3).
\end{aligned} \tag{13}$$

A total of 6 features (F13–F18) is extracted from each of the handwritten digit images for the present work.

**3.5. The Legendre Moment.** The 2D Legendre moment [29] of order  $(p+q)$  of an object with intensity function  $f(x, y)$  is defined as follows:

$$\begin{aligned}
L_{pq} &= \frac{(2p+1)(2q+1)}{4} \\
&\quad \cdot \iint_{-1}^{+1} P_p(x) P_q(y) f(x, y) dx dy,
\end{aligned} \tag{14}$$

where the kernel function  $P_p(x)$  denotes the  $p$ th-order Legendre polynomial and is given by

$$P_p(x) = \sum_{k=0}^p C_{pk} [(1-x)^k + (-1)^p(1+x)^k], \tag{15}$$

where

$$C_{pk} = \frac{(-1)^k}{2^{k+1}} \frac{(p+k)!}{(p-k)!(k!)^2}. \tag{16}$$

Since the Legendre polynomials are orthogonal over the interval  $[-1, 1]$  [20], a square image of  $N \times N$  pixels with intensity function  $f(i, j)$ , with  $1 \leq i, j \leq N$ , must be scaled to be within the region  $-1 \leq x, y \leq 1$ . The graphical plot for first 10 Legendre polynomials is shown in Figures 2(a)-2(b). When an analog image is digitized to its discrete form, the 2D Legendre moments  $L_{pq}$ , defined by (14), is usually approximated by the formula:

$$\begin{aligned}
L_{pq} &= \frac{(2p+1)(2q+1)}{(N-1)^2} \sum_{i=1}^N \sum_{j=1}^N P_p(x_i) P_q(y_j) f(x_i, y_j),
\end{aligned} \tag{17}$$

where  $x_i = (2i-N-1)/(N-1)$  and  $y_j = (2j-N-1)/(N-1)$ , and, for a binary image,  $f(x_i, y_j)$  is given as

$$f(x_i, y_j) = \begin{cases} 1, & \text{if } (i, j) \text{ is in original object,} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

As indicated by Liao and Pawlak [30], (17) is not a very accurate approximation of (14). For achieving better accuracy, they proposed to use the following approximated form:

$$\tilde{L}_{pq} = \frac{(2p+1)(2q+1)}{4} \sum_{i=1}^N \sum_{j=1}^N h_{pq}(x_i, y_j) f(x_i, y_j), \tag{19}$$

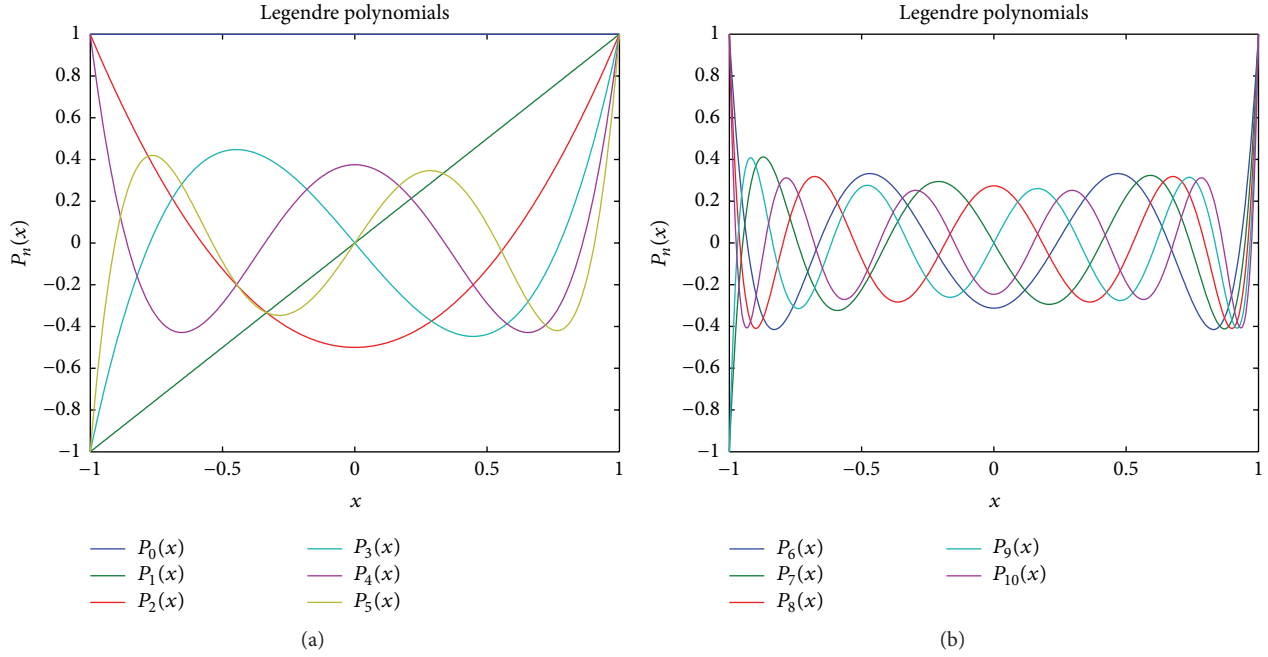


FIGURE 2: Graph showing the plots for the two-dimensional Legendre polynomials  $P_n(x)$ : (a)  $P_1(x)$  to  $P_5(x)$  and (b)  $P_6(x)$  to  $P_{10}(x)$ .

where

$$h_{pq}(x_i, y_j) = \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} \int_{y_j - \Delta y/2}^{y_j + \Delta y/2} P_p(x) P_q(y) dx dy \quad (20)$$

with  $\Delta x = x_i - x_{i-1} = 2/(N-1)$  and  $\Delta y = y_j - y_{j-1} = 2/(N-1)$ .

To evaluate the double integral  $h_{pq}(x_i, y_j)$  defined by (20), an alternative extended Simpson rule was proposed by Liao and Pawlak. These values were then used to calculate the 2D Legendre moments  $\tilde{L}_{pq}$  defined by (19). Therefore, this method requires a large number of computing operations. As one can see,  $\tilde{L}_{pq}$  can be expressed with the help of a useful formula that will be given below as a linear combination of  $L_{mn}$ , with  $0 \leq m \leq p$ ,  $0 \leq n \leq q$ .

A set of 10 Legendre moments (F19–F28) can also be derived based on the set of *invariant moments* found in the previous subsection:

$$L_{00} = \phi_{00},$$

$$L_{10} = \left(\frac{3}{4}\right) \phi_{10},$$

$$L_{01} = \left(\frac{3}{4}\right) \phi_{01},$$

$$L_{20} = \left(\frac{5}{4}\right) \left[ \left(\frac{3}{2}\right) \phi_{20} - \left(\frac{1}{2}\right) \phi_{00} \right],$$

$$L_{02} = \left(\frac{5}{4}\right) \left[ \left(\frac{3}{2}\right) \phi_{02} - \left(\frac{1}{2}\right) \phi_{00} \right],$$

$$L_{11} = \left(\frac{9}{4}\right) \phi_{11},$$

$$L_{30} = \left(\frac{7}{4}\right) \left[ \left(\frac{5}{2}\right) \phi_{30} - \left(\frac{3}{2}\right) \phi_{10} \right],$$

$$L_{03} = \left(\frac{7}{4}\right) \left[ \left(\frac{5}{2}\right) \phi_{03} - \left(\frac{3}{2}\right) \phi_{01} \right],$$

$$L_{21} = \left(\frac{15}{4}\right) \left[ \left(\frac{3}{2}\right) \phi_{21} - \left(\frac{1}{2}\right) \phi_{01} \right],$$

$$L_{12} = \left(\frac{15}{4}\right) \left[ \left(\frac{3}{2}\right) \phi_{12} - \left(\frac{1}{2}\right) \phi_{10} \right].$$

(21)

**3.6. Zernike Moments.** Zernike polynomials are orthogonal series of basis functions normalized over a unit circle. The complexity of these polynomials increases with increasing polynomial order [31]. To calculate the Zernike moments, the image (or region of interest) is first mapped to the unit disc using polar coordinates, where the center of the image is the origin of the unit disc. The pixels falling outside the unit disc are not considered here. The coordinates are then described by the length of the vector from the origin to the coordinate point. The mapping from Cartesian to polar coordinates is defined as follows:

$$x = r \cos \theta,$$

$$y = r \sin \theta,$$

(22)

where

$$\begin{aligned} r &= \sqrt{x^2 + y^2}, \\ \theta &= \tan^{-1} \left( \frac{y}{x} \right). \end{aligned} \quad (23)$$

An important attribute of the geometric representations of Zernike polynomials is that lower order polynomials approximate the global features of the shape/surface, while the higher ordered polynomials capture local shape/surface features. Zernike moments are a class of orthogonal moments and have been shown to be effective in terms of image representation.

Zernike introduced a set of complex polynomials which forms a complete orthogonal set over the interior of the unit circle; that is,  $x^2 + y^2 = 1$ . Let the set of these polynomials be denoted by  $\{V_{nm}(x, y)\}$ . The form of these polynomials is as follows:

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta), \quad (24)$$

where

$n$ : positive integer or zero,

$m$ : positive and negative integers subject to constraints  $n - |m|$  even,  $|m| \leq n$ ,

$\rho$ : length of vector from origin to  $f(x, y)$  pixel,

$\theta$ : angle between vector  $\rho$  and  $x$ -axis in counterclockwise direction.

As mentioned above, the complex Zernike moments of order  $n$  with repetition  $m$  for a continuous image function  $f(x, y)$  are defined as follows:

$$V_{nm} = \frac{n+1}{\pi} \iint f(x, y) V_{nm}^*(\rho, \theta) dx dy \quad (25)$$

in the  $xy$  image plane where  $x^2 + y^2 \leq 1$  and  $*$  indicates the complex conjugate. Note that, for the moments to be orthogonal, the image must be scaled within a unit circle centered at the origin and

$$\begin{aligned} V_{nm} &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta) R_{nm}(\rho) \exp(-jm\theta) \rho d\rho d\theta \end{aligned} \quad (26)$$

in polar coordinates. The Zernike moment of the rotated image in the same coordinates is given by

$$\begin{aligned} V_{nm}^r &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta - \alpha) R_{nm}(\rho) \exp(-jm\theta) \rho d\rho d\theta. \end{aligned} \quad (27)$$

By change of variable,  $\theta_1 = \theta - \alpha$ ,

$$\begin{aligned} V_{nm}^r &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta_1) R_{nm}(\rho) \\ &\cdot \exp(-jm(\theta_1 + \alpha)) \rho d\rho d\theta = \left[ \frac{n+1}{\pi} \right. \\ &\cdot \left. \int_0^{2\pi} \int_0^1 f(\rho, \theta_1) R_{nm}(\rho) \exp(-jm\theta_1) \rho d\rho d\theta \right] \\ &\cdot \exp(-jm\alpha) = V_{nm} \exp(-jm\alpha). \end{aligned} \quad (28)$$

Equation (28) shows that Zernike moments have simple rotational transformation properties; each Zernike moment merely acquires a phase shift on rotation. This simple property leads to the conclusion that the magnitudes of the Zernike moments of a rotated image function remain identical to those before rotation. Thus, the magnitude of the Zernike moment,  $|V_{nm}|$ , can be taken as a rotation invariant feature of the underlying image function. The real-valued radial polynomial  $R_{nm}(\rho)$  is defined as follows:

$$\begin{aligned} R_{nm}(\rho) &= \sum_{s=0}^{(n-|m|)/2} (-1)^s \\ &\cdot \frac{(n-s)!}{s! ((n+|m|)/2 - s)! ((n-|m|)/2 - s)!} \rho^{n-2s}, \end{aligned} \quad (29)$$

where  $n - |m| = \text{even}$  and  $|m| \leq n$ .

Zernike moments may also be derived from conventional moments  $\mu_{pq}$  as follows:

$$\begin{aligned} Z_{nl} &= \frac{(n+1)}{\pi} \\ &\cdot \sum_{k=l}^n \sum_{j=0}^q \sum_{m=0}^l (-i)^m \binom{q}{j} \binom{l}{m} B_{nlk} \mu_{k-2j-l+m, 2j+l-m}. \end{aligned} \quad (30)$$

Zernike moments may be more easily derived from rotational moments,  $D_{nk}$ , by

$$Z_{nl} = \sum_{k=l}^n B_{nlk} D_{nk}. \quad (31)$$

When computing the Zernike moments, if the center of a pixel falls inside the border of unit disk  $x^2 + y^2 \leq 1$ , this pixel will be used in the computation; otherwise, the pixel will be discarded. Therefore, the area covered by the moment computation is not exactly the area of the unit disk. Advantages of Zernike moments can be summarized as follows:

- (1) The magnitude of Zernike moment has rotational invariant property.
- (2) They are robust to noise and shape variations to some extent.
- (3) Since the basis is orthogonal, they have minimum redundant information.

TABLE 1: List of Zernike moments and their corresponding numbers of features from order 0 to order 10.

Order ( $n$ )	Zernike moments of order $n$ with repetition $m$ ( $V_{nm}$ )	Total number of moments up to order 10
0	$V_{00}$	
1	$V_{11}$	
2	$V_{20}, V_{22}$	
3	$V_{31}, V_{33}$	
4	$V_{40}, V_{42}, V_{44}$	
5	$V_{51}, V_{53}, V_{55}$	36
6	$V_{60}, V_{62}, V_{64}, V_{66}$	
7	$V_{71}, V_{73}, V_{75}, V_{77}$	
8	$V_{80}, V_{82}, V_{84}, V_{86}, V_{88}$	
9	$V_{91}, V_{93}, V_{95}, V_{97}, V_{99}$	
10	$V_{10,0}, V_{10,2}, V_{10,4}, V_{10,6}, V_{10,8}, V_{10,10}$	

- (4) An image can better be described by a small set of its Zernike moments than any other types of moments such as geometric moments.
- (5) A relatively small set of Zernike moments can characterize the global shape of pattern. Lower order moments represent the global shape of pattern whereas the higher order moments represent the details.

Therefore, we choose Zernike moments as our shape descriptor in digit recognition process. Table 1 lists the rotation invariant Zernike moment features (F29–F64) and their corresponding numbers from order 0 to order 10 used for the present work.

The defined features on the Zernike moments are only rotation invariant. To obtain scale and translation invariance, the digit image is first subjected to a normalization process using its regular moments. The rotation invariant Zernike features are then extracted from the scale and translation normalized image.

**3.7. Complex Moments.** The notion of complex moments was introduced in [32] as a simple and straightforward technique to derive a set of invariant moments. The two-dimensional complex moments of order  $(p, q)$  for the image function  $f(x, y)$  are defined by

$$C_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} (x + jy)^p (x - jy)^q f(x, y) dx dy, \quad (32)$$

where  $p$  and  $q$  are nonnegative integers and  $j = \sqrt{-1}$ . Some advantages of the complex moments can be described as follows:

- (1) When the central complex moments are taken as the features, the effects of the image's lateral displacement can be eliminated.

- (2) A set of complex moment invariants can also be derived which are invariant to the rotation of the object.
- (3) Since the complex moment is an intermediate step between ordinary moments and moment invariants, it is relatively more simple to compute and more powerful than other moment features in any pattern classification problem.

The complex moments of order  $(p, q)$  are a linear combination with complex coefficients of all the geometric moments  $\{M_{nm}\}$  satisfying  $p + q = n + m$ . In polar coordinates, the complex moments of order  $(p + q)$  can be written as follows:

$$C_n^l = C_{pq} = \int_0^{2\pi} \int_0^{+\infty} \rho^{p+q} e^{j(p-q)\theta} f(\rho \cos \theta, \rho \sin \theta) \rho d\rho d\theta, \quad (33)$$

where  $p+q = n$  and  $p-q = l$  denote the order and repetition of the complex moments, respectively. If the complex moment of the original image and that of the rotated image in the same polar coordinates are denoted by  $C_{pq}$  and  $C_{pq}^r$ , the relationship [33] between them is given as follows:

$$C_{pq}^r = C_{pq} e^{-j(p-q)\theta}, \quad (34)$$

where  $\theta$  is the angle at which the original image is rotated. The complex moment features represent the invariant properties to lateral displacement and rotation. Based on the definition of moment invariants, we know that as the image is rotated, each complex moment goes through all possible phases of a complex number while its magnitude  $|C_{pq}|$  remains unchanged. If the exponential factor of the complex moment is canceled out, we will obtain its absolute invariant value, which is invariant to the rotation of the images. The rotation invariant complex moment features (F65–F130) and their corresponding numbers from order 0 to order 10 used for the present work are listed in Table 2.

Finally, a feature vector consisting of 130 moment based features is calculated from each of the handwritten numeral images belonging to five different scripts. Summarization of the overall moment based feature set used in the present work is enlisted in Table 3.

## 4. Experimental Study and Analysis

In this section, we present the detailed experimental results to illustrate the suitability of moment based approach to handwritten digit recognition. All the experiments are implemented in MATLAB 2010 under a Windows XP environment on an Intel Core2 Duo 2.4 GHz processor with 1 GB of RAM and performed on gray-scale digit images. The accuracy, used as assessment criteria for measuring the performance of the proposed system, is expressed as follows:

$$\text{Accuracy Rate (\%)} = \frac{\text{Number of Correctly classified digits}}{\text{Total number of digits}} \times 100\%. \quad (35)$$



TABLE 2: List of complex moments and their corresponding numbers of features from order 0 to order 10.

Order (n)	Complex moments ( $C_{pq}$ )	Complex moments of order n with repetition l ( $C_n^l$ )	Total number of moments up to order 10
0	$C_{00}$	$C_0^0$	
1	$C_{10}, C_{01}$	$C_1^1, C_1^{-1}$	
2	$C_{20}, C_{11}, C_{02}$	$C_2^2, C_2^0, C_2^{-2}$	
3	$C_{30}, C_{21}, C_{12}, C_{03}$	$C_3^3, C_3^1, C_3^{-1}, C_3^{-3}$	
4	$C_{40}, C_{31}, C_{22}, C_{13}, C_{04}$	$C_4^4, C_4^2, C_4^0, C_4^{-2}, C_4^{-4}$	
5	$C_{50}, C_{41}, C_{32}, C_{23}, C_{14}, C_{05}$	$C_5^5, C_5^3, C_5^1, C_5^{-1}, C_5^{-3}, C_5^{-5}$	66
6	$C_{60}, C_{51}, C_{42}, C_{33}, C_{24}, C_{15}, C_{06}$	$C_6^6, C_6^4, C_6^2, C_6^0, C_6^{-2}, C_6^{-4}, C_6^{-6}$	
7	$C_{70}, C_{61}, C_{52}, C_{43}, C_{34}, C_{25}, C_{16}, C_{07}$	$C_7^7, C_7^5, C_7^3, C_7^1, C_7^{-1}, C_7^{-3}, C_7^{-4}, C_7^{-7}$	
8	$C_{80}, C_{71}, C_{62}, C_{53}, C_{44}, C_{35}, C_{26}, C_{17}, C_{08}$	$C_8^8, C_8^6, C_8^4, C_8^2, C_8^0, C_8^{-2}, C_8^{-4}, C_8^{-6}, C_8^{-8}$	
9	$C_{90}, C_{81}, C_{72}, C_{63}, C_{54}, C_{45}, C_{36}, C_{27}, C_{18}, C_{09}$	$C_9^9, C_9^7, C_9^5, C_9^3, C_9^1, C_9^{-1}, C_9^{-3}, C_9^{-5}, C_9^{-7}, C_9^{-9}$	
10	$C_{10,0}, C_{9,1}, C_{8,2}, C_{7,3}, C_{6,4}, C_{5,5}, C_{4,6}, C_{3,7}, C_{2,8}, C_{1,9}, C_{0,10}$	$C_{10}^{10}, C_{10}^8, C_{10}^6, C_{10}^4, C_{10}^2, C_{10}^0, C_{10}^{-2}, C_{10}^{-4}, C_{10}^{-6}, C_{10}^{-8}, C_{10}^{-10}$	

TABLE 3: Description of feature vector.

Serial number	List of moments	Number of features
1	Geometric moment (F1-F5)	5
2	Moment invariant (F6-F12)	7
3	Affine moment invariant (F13-F18)	6
4	Legendre moment (F19-F28)	10
5	Zernike moment (F29-F64)	36
6	Complex moment (F65-F130)	66
	Total	<b>130</b>

4.1. Detailed Dataset Description. Handwritten numerals from five different popular scripts, namely, *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu*, are used in the experiments for investigating the effectiveness of the moment based feature sets as compared to conventional features. *Indo-Arabic* or *Eastern-Arabic* is widely used in the Middle-East and also in the Indian subcontinent. On the other hand, *Devanagari* and *Bangla* are ranked as the top two popular (in terms of the number of native speakers) scripts in the Indian subcontinent [34]. *Roman*, originally evolved from the *Greek* alphabet, is spoken and used all over the world. Also, *Telugu*, one of the oldest and popular South Indian languages of India, is spoken by more than 74 million people [34]. It essentially ranks third by the number of native speakers in India.

The present approach is tested on the database named as *CMATERdb3*, where *CMATER* stands for Center for Micro-processor Application for Training Education and Research, a research laboratory at Computer Science and Engineering Department of Jadavpur University, India, where the current research activity took place. *db* stands for *database*, and the numeric value 3 represents *handwritten digit recognition* database stored in the said database repository. The testing is currently done on four versions of *CMATERdb3*, namely, *CMATERdb3.1.1*, *CMATERdb3.2.1*, *CMATERdb3.3.1*, and *CMATERdb3.4.1* representing the databases created for handwritten digit recognition system for four major scripts, namely, *Bangla*, *Devanagari*, *Indo-Arabic*, and *Telugu*, respectively.

Each of the digit images are first preprocessed using basic operations of skew corrections and morphological filtering [25] and then binarized using an adaptive global threshold value computed as the average of minimum and maximum intensities in that image. The binarized digit images may contain noisy pixels which have been removed by using Gaussian filter [25]. A well-known algorithm known as Canny Edge Detection algorithm [25] is then applied for smoothing the edges of the binarized digit images. Finally, the bounding rectangular box of each digit image is separately normalized to  $32 \times 32$  pixels. Database is made available *freely* in the *CMATER* website (<http://www.cmaterju.org/cmaterdb.htm>) and at <http://code.google.com/p/cmaterdb/>.

A dataset of 3000 digit samples is considered for each of the *Devanagari*, *Indo-Arabic*, and *Telugu* scripts. For each of these datasets, 2000 samples are used for training purpose and the rest of the samples are used for the test purpose, whereas a dataset of 6000 samples is used by selecting 600 samples for each of 10-digit classes of handwritten *Bangla* digits. A training set of 4000 samples and a test set of 2000 samples are then chosen for *Bangla* numerals by considering equal number of digit samples from each class. For *Roman* numerals, a dataset of 6000 training samples is formed by random selection from the standard handwritten MNIST [7] training dataset of size 60,000 samples. In the same way, 4000 digit samples are selected from MNIST test dataset of size 10,000 samples. These digit samples are enclosed in a minimum bounding square and are normalized to  $32 \times 32$  pixels dimension. Typical handwritten digit samples taken from the abovementioned databases used for evaluating the present work are shown in Figure 3.

4.2. Recognition Process. To realize the effectiveness of the proposed approach, our comprehensive experimental tests are conducted on the five aforementioned datasets. A total of 6,000 (for *Devanagari*, *Indo-Arabic*, and *Telugu* scripts) numerals have been used for the training purpose whereas the remaining 3000 numerals (1000 from each of the script) have been used for the testing purpose. For *Bangla* and *Roman* scripts, a total of 8,000 numerals (4000 taken from

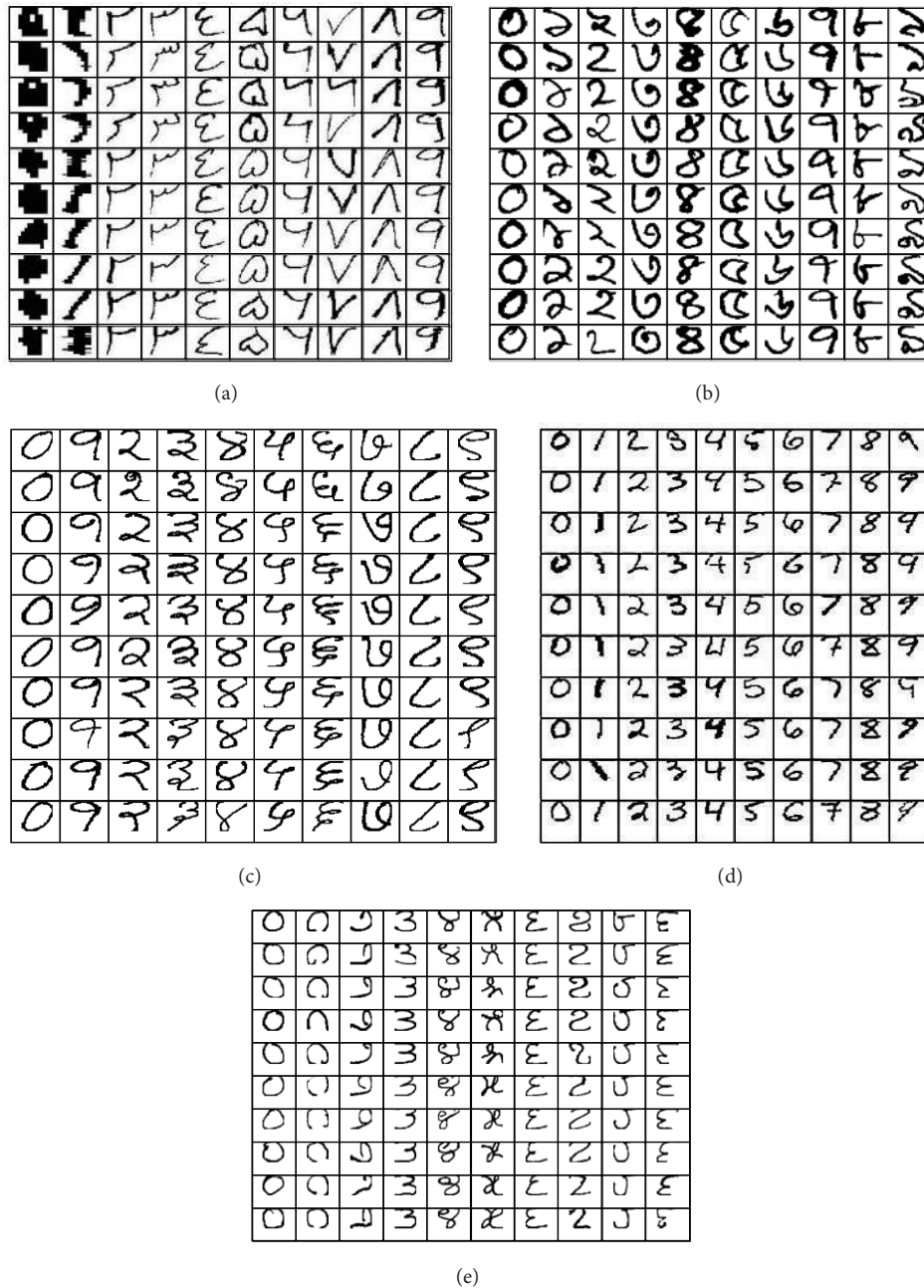


FIGURE 3: Samples of digit images taken from CMATER and MNIST databases written in five different scripts: (a) *Indo-Arabic*, (b) *Bangla*, (c) *Devanagari*, (d) *Roman*, and (e) *Telugu*.

each script) have been used for the training purpose whereas the remaining 4,000 numerals (2000 taken from each script) have been used for the testing purpose. The designed feature set has been individually applied to eight well-known classifiers, namely, Naïve Bayes, Bayes Net, MLP, SVM, Random Forest, Bagging, Multiclass Classifier, and Logistic. For the present work, the following abovementioned classifiers with the given parameters are designed:

Naïve Bayes: Naïve Bayes classifier: for details, refer to [35].

Bayes Net: Estimator = SimpleEstimator-A 0.5, search algorithm = K2.

MLP: Learning Rate = 0.3, Momentum = 0.2, Number of Epochs = 1000, minerror = 0.02.

SVM: Support Vector Machine using radial basis kernel with ( $p = 1$ ): for details, refer to [36].

Random Forest: Ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees: for details, refer to [37].

TABLE 4: Recognition accuracies of eight classifiers and their corresponding ranks using 12 different datasets (ranks in the parentheses are used for performing the Friedman test).

Datasets	Recognition accuracy (%)							
	Classifiers							
	Naïve Bayes	Bayes Net	MLP	SVM	Random Forest	Bagging	Multiclass Classifier	Logistic
#1	86 (8)	92 (7)	100 (1)	99 (2.5)	97 (6)	99 (2.5)	98 (4.5)	98 (4.5)
#2	99 (3)	98 (5.5)	100 (1)	99 (3)	96 (7.5)	96 (7.5)	97 (5.5)	99 (3)
#3	98 (5)	94 (7)	100 (1)	93 (8)	99 (2.5)	98 (5)	99 (2.5)	98 (5)
#4	93 (8)	94 (7)	99 (1.5)	98 (3.5)	98 (3.5)	97 (5)	96 (6)	99 (1.5)
#5	91 (8)	92 (7)	100 (1.5)	99 (3.5)	99 (3.5)	97 (6)	98 (5)	100 (1.5)
#6	99 (2.5)	98 (4.5)	100 (1)	96 (7)	97 (6)	98 (4.5)	99 (2.5)	95 (8)
#7	91 (8)	94 (7)	100 (1)	99 (3)	99 (3)	99 (3)	98 (5.5)	98 (5.5)
#8	91 (8)	96 (7)	100 (1)	98 (4.5)	98 (4.5)	97 (6)	99 (2.5)	99 (2.5)
#9	92 (8)	94 (7)	100 (1.5)	100 (1.5)	97 (6)	99 (4)	99 (4)	99 (4)
#10	99 (2.5)	97 (6.5)	100 (1)	99 (2.5)	97 (6.5)	97 (6.5)	98 (4)	97 (6.5)
#11	94 (8)	96 (7)	100 (1)	98 (5)	99 (3)	98 (5)	99 (3)	99 (3)
#12	98 (4)	98 (4)	100 (1)	97 (7)	98 (4)	99 (2)	97 (7)	97 (7)
Mean rank	<b>R<sub>1</sub> = 6.08</b>	<b>R<sub>2</sub> = 6.37</b>	<b>R<sub>3</sub> = 1.125</b>	<b>R<sub>4</sub> = 4.25</b>	<b>R<sub>5</sub> = 4.67</b>	<b>R<sub>6</sub> = 4.75</b>	<b>R<sub>7</sub> = 4.33</b>	<b>R<sub>8</sub> = 4.33</b>

Bagging: Bagging Classifier: for detail, refer to [38].

Multiclass Classifier: Method = "1 against all," randomWidthFactor = 2.0, seed = 1.

Logistic: LogitBoost is used with simple regression functions as base learner: for details, refer to [39].

The design parameters of classifiers are chosen as typical values used in the literature or by experience. The classifiers are not specifically tuned for the dataset at hand even though they may achieve a better performance with another parameter set, since the goal is to design an automated handwritten digit recognition system based on the chosen set of classifiers.

The digit recognition performances of the present technique using each of these classifiers and their corresponding success rates achieved at 95% confidence level are shown in Figures 4(a)-4(b), respectively. It can be seen from Figure 4 that the highest digit recognition accuracy has been achieved by the MLP classifier which are found to be 99.3%, 99.5%, 98.92%, 99.77%, and 98.8% on *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu* scripts, respectively. The performance analysis involves two parameters, namely, Model Building Time (MBT) and Recognition Time (RT). MBT is based on the time required to train the system on the given training samples whereas RT is based on the time required to recognize the given test samples. The MBT and RT required for the abovementioned classifiers on all the five databases are shown in Figures 5(a)-5(b).

**4.3. Statistical Significance Tests.** The statistical significance test is one of the essential ways for validating the performance of the multiple classifiers using multiple datasets. To do so, we have performed a safe and robust nonparametric Friedman test [40] with the corresponding post hoc tests on *Indo-Arabic* script database. For the present experimental setup, the number of datasets ( $N$ ) and the number of classifiers ( $k$ ) are set as 12 and 8, respectively. These datasets are chosen randomly from the test set. The performances of the classifiers

on different datasets are shown in Table 4. On the basis of these performances, the classifiers are then ranked for each dataset separately, the best performing algorithm gets the rank 1, the second best gets rank 2, and so on (see Table 4). In case of ties, average ranks are assigned to the classifiers to break the tie.

Let  $r_j^i$  be the rank of the  $j$ th classifier on  $i$ th dataset. Then, the mean of the ranks of the  $j$ th classifier over all the  $N$  datasets will be computed as follows:

$$R_j = \frac{1}{N} \sum_{i=1}^N r_j^i. \quad (36)$$

The null hypothesis states that all the classifiers are equivalent and so their ranks  $R_j$  should be equal. To justify it, the Friedman statistic [40] is computed as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]. \quad (37)$$

Under the current experimentation, this statistic is distributed according to  $\chi_F^2$  with  $k-1$  ( $=7$ ) degrees of freedom. Using (37), the value of  $\chi_F^2$  is calculated as 30.46. From the table of critical values (see any standard statistical book), the value of  $\chi_F^2$  with 7 degrees of freedom is 14.0671 for  $\alpha = 0.05$  (where  $\alpha$  is known as level of significance). It can be seen that the computed  $\chi_F^2$  differs significantly from the standard  $\chi_F^2$ . So, the null hypothesis is rejected.

Singh et al. [40] derived a better statistic using the following formula:

$$F_F = \frac{(N-1) \chi_F^2}{N(k-1) - \chi_F^2}. \quad (38)$$

$F_F$  is distributed according to the  $F$ -distribution with  $k-1$  ( $=7$ ) and  $(k-1)(N-1)$  ( $=77$ ) degrees of freedom. Using (38), the value of  $F_F$  is calculated as 8.0659. The critical value of  $F$  ( $7, 77$ ) for  $\alpha = 0.05$  is 2.147 (see any standard statistical book)

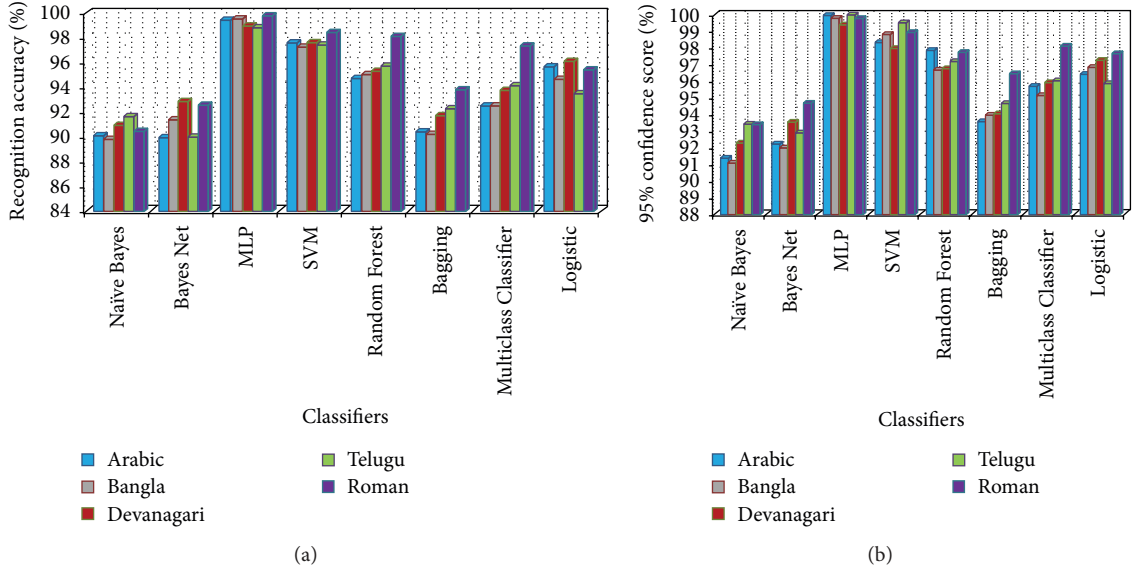


FIGURE 4: Graph showing (a) recognition accuracies and (b) 95% confidence scores of the proposed handwritten digit recognition technique using eight well-known classifiers on digits of five different scripts.

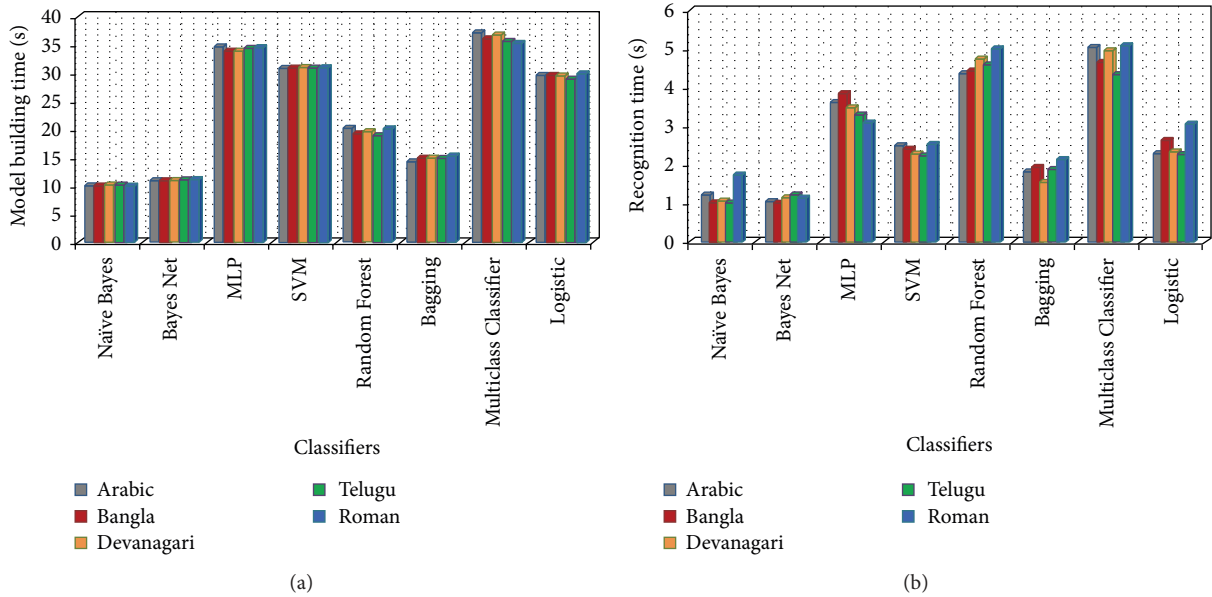


FIGURE 5: Graphical comparison of (a) MBTs and (b) RTs required by eight different classifiers on all the five databases for handwritten digit recognition.

which shows a significant difference between the standard and calculated values of  $F_p$ . Thus, both Friedman and Iman et al. statistics reject the null hypothesis.

As the null hypothesis is rejected, a post hoc test known as the Nemenyi test [40] is carried out for pairwise comparisons of the best and worst performing classifiers. The performances of two classifiers are significantly different if the corresponding average ranks differ by at least the critical difference (CD) which is expressed as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \quad (39)$$

For the Nemenyi test, the value of  $q_{0.05}$  for eight classifiers is 3.031 (see Table 5(a) of [41]). So, the CD is calculated as  $3.031 \sqrt{8.9/6.12}$ , that is, 3.031, using (39). Since the difference between mean ranks of the best and worst classifier is much greater than the CD (see Table 3), we can conclude that there is a significant difference between the performing abilities of the classifiers. For comparing all classifiers with a *control classifier* (say MLP), we have applied the Bonferroni-Dunn test [40]. For this test, CD is calculated using the same (39). But here, the value of  $q_{0.05}$  for eight classifiers is 2.690 (see Table 5(b) of [41]). So, the CD for the Bonferroni-Dunn test is calculated as  $2.690 \sqrt{8.9/6.12}$ , that is, 2.690. As the

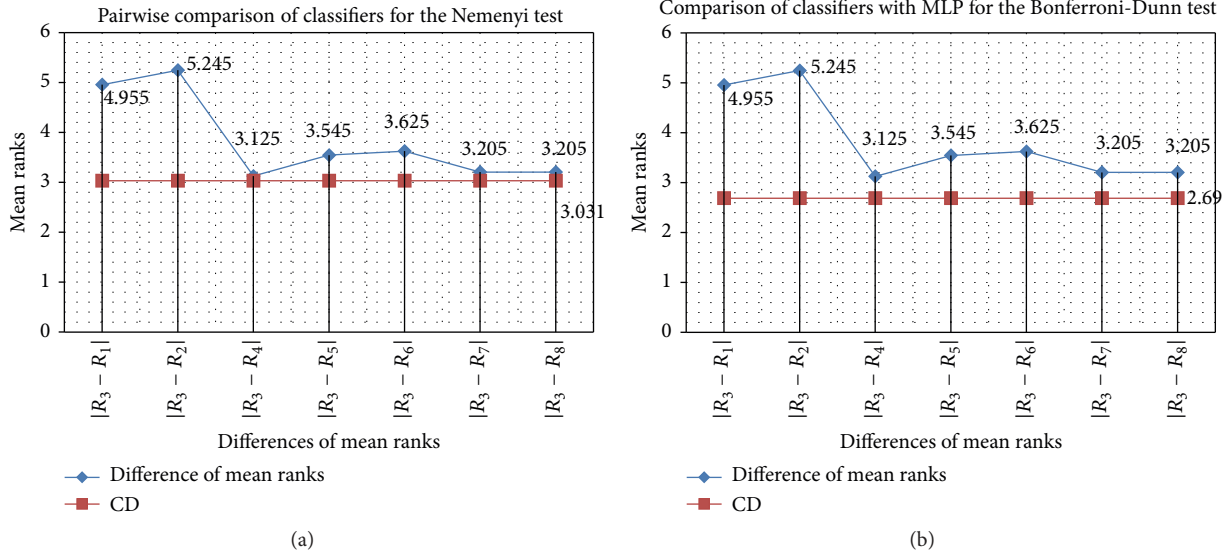


FIGURE 6: Graphical representation of comparison of multiple classifiers for (a) the Nemenyi test and (b) the Bonferroni-Dunn test.

difference between the mean ranks of any classifier and MLP is always greater than CD (see Table 3), the chosen control classifier performs significantly better than other classifiers for *Indo-Arabic* database. A graphical representation of the abovementioned post hoc tests for comparison of eight different classifiers on Dataset #1 is shown in Figure 6. Similarly, it can also be shown for *Bangla*, *Devanagari*, *Roman*, and *Telugu* databases that the chosen classifier (MLP) performs significantly better than the other seven classifiers.

4.4. *Comparison among Moment Based Features.* For the justification of the feature set used in the present work, the diverse combinations of six different types of moments, namely, geometric moment ( $F1-F5$ ), moment invariant ( $F6-F12$ ), affine moment invariant ( $F13-F18$ ), Legendre moment ( $F19-F28$ ), Zernike moment ( $F29-F64$ ), and complex moment ( $F65-F130$ ), are compared by considering all the possible combinations. This is done for measuring the discriminating strength of the individual moment features and their combinations based on their complementary information. These can be listed as follows:

- (a) Geometric moment + moment invariant + affine Moment invariant ( $F1-F18$ ).
- (b) Legendre moment ( $F19-F28$ ).
- (c) Geometric moment + moment invariant + affine moment invariant + Legendre moment ( $F1-F28$ ).
- (d) Zernike moment ( $F29-F64$ ).
- (e) Geometric moment + moment invariant + affine moment invariant + Legendre moment + Zernike moment ( $F1-F64$ ).
- (f) Legendre moment + Zernike moment ( $F19-F64$ ).
- (g) Complex moment ( $F65-F130$ ).
- (h) Zernike moment + complex moment ( $F29-F130$ ).

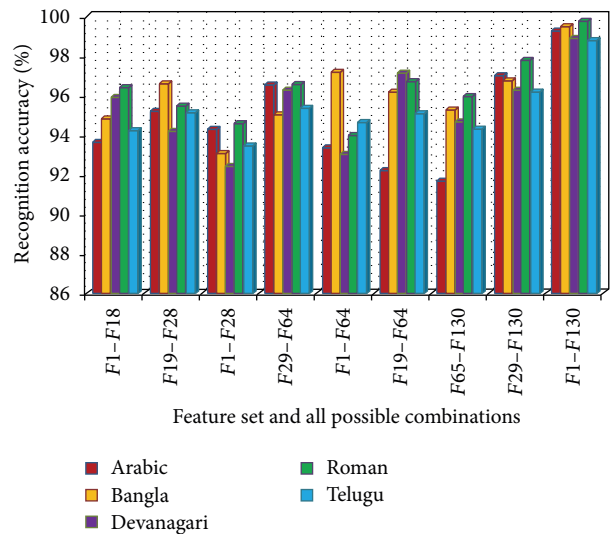


FIGURE 7: Graphical comparison showing the recognition accuracies of all the possible combinations of moment based features achieved by MLP classifier.

- (i) Geometric moment + moment invariant + affine moment invariant + Legendre moment + Zernike moment + complex moment ( $F1-F130$ ).

The graphical comparison of the corresponding numeral recognition accuracies achieved by MLP classifier over the same test set is shown in Figure 7. It can be observed from Figure 7 that the present combination of moment feature set outperforms all the other possible combinations.

4.5. *Detail Evaluation of MLP Classifier.* In the present work, detailed error analysis with respect to different parameters, namely, Kappa statistics, mean absolute error (MAE), root

TABLE 5: Statistical performance measures along with their respective means (styled in bold) achieved by the proposed technique on handwritten *Indo-Arabic* numerals (here, MAE means mean absolute error, RMSE means root mean square error, TPR means True Positive rate, FPR means False Positive rate, MCC means Matthews Correlation Coefficient, and AUC means Area under ROC).

Class	Statistical performance measures									
	Kappa statistics	MAE	RMSE	TPR	FPR	Precision	Recall	<i>F</i> -measure	MCC	AUC
"0"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"1"				0.990	0.001	0.990	0.990	0.990	0.989	1.000
"2"				0.960	0.002	0.980	0.960	0.970	0.966	0.990
"3"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"4"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"5"	0.9922	0.1758	0.293	1.000	0.000	1.000	1.000	1.000	1.000	1.000
"6"				0.990	0.004	0.961	0.990	0.975	0.973	0.999
"7"				0.990	0.000	1.000	0.990	0.995	0.994	1.000
"8"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"9"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
Mean	<b>0.9922</b>	<b>0.1758</b>	<b>0.293</b>	<b>0.993</b>	<b>0.0007</b>	<b>0.9931</b>	<b>0.993</b>	<b>0.993</b>	<b>0.9922</b>	<b>0.9989</b>

TABLE 6: Statistical performance measures along with their respective means (styled in bold) achieved by the proposed technique on handwritten *Bangla* numerals.

Class	Statistical performance measures									
	Kappa statistics	MAE	RMSE	TPR	FPR	Precision	Recall	<i>F</i> -measure	MCC	AUC
"0"				1.000	0.001	0.990	1.000	0.995	0.994	1.000
"1"				1.000	0.002	0.980	1.000	0.990	0.989	1.000
"2"				1.000	0.001	0.990	1.000	0.995	0.994	1.000
"3"				0.990	0.000	1.000	0.990	0.995	0.994	1.000
"4"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"5"	0.9944	0.0535	0.115	0.990	0.001	0.990	0.990	0.990	0.989	1.000
"6"				0.970	0.000	1.000	0.970	0.985	0.983	1.000
"7"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"8"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"9"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
Mean	<b>0.9944</b>	<b>0.0535</b>	<b>0.115</b>	<b>0.995</b>	<b>0.0005</b>	<b>0.995</b>	<b>0.995</b>	<b>0.995</b>	<b>0.9943</b>	<b>1.000</b>

TABLE 7: Statistical performance measures along with their respective means (styled in bold) achieved by the proposed technique on handwritten *Devanagari* numerals.

Class	Statistical performance measures									
	Kappa statistics	MAE	RMSE	TPR	FPR	Precision	Recall	<i>F</i> -measure	MCC	AUC
"0"				0.969	0.002	0.984	0.969	0.977	0.974	0.999
"1"				0.969	0.000	1.000	0.969	0.984	0.983	1.000
"2"				1.000	0.005	0.956	1.000	0.977	0.975	1.000
"3"				0.985	0.003	0.970	0.985	0.977	0.975	0.999
"4"				1.000	0.002	0.985	1.000	0.992	0.992	1.000
"5"	0.988	0.0342	0.0847	0.985	0.000	1.000	0.985	0.992	0.991	1.000
"6"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"7"				0.985	0.000	1.000	0.985	0.992	0.991	1.000
"8"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"9"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
Mean	<b>0.988</b>	<b>0.0342</b>	<b>0.0857</b>	<b>0.9893</b>	<b>0.0012</b>	<b>0.9895</b>	<b>0.9893</b>	<b>0.9891</b>	<b>0.9881</b>	<b>0.9998</b>

mean square error (RMSE), True Positive rate (TPR), False Positive rate (FPR), precision, recall, *F*-measure, Matthews Correlation Coefficient (MCC), and Area under ROC (AUC), is computed. Tables 5–9 provide the said statistical measurements for handwritten numeral recognition written in *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu* scripts, respectively.

## 5. Conclusion

India is a multilingual and multiscript country comprising of 12 different scripts. But there are not much competent works done towards handwritten numeral recognition of *Indic* scripts. The following issues are observed with handwritten digit recognition system: (1) mostly they have worked

TABLE 8: Statistical performance measures along with their respective means (styled in bold) achieved by the proposed technique on handwritten *Roman* numerals.

Class	Statistical performance measures									
	Kappa statistics	MAE	RMSE	TPR	FPR	Precision	Recall	<i>F</i> -measure	MCC	AUC
"0"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"1"				0.995	0.001	0.995	0.995	0.995	0.994	0.999
"2"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"3"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"4"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"5"	0.9975	0.16	0.2716	0.995	0.000	1.000	0.995	0.997	0.997	1.000
"6"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"7"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"8"				0.994	0.001	0.989	0.994	0.991	0.990	0.999
"9"				0.994	0.001	0.994	0.994	0.994	0.994	0.999
Mean	<b>0.9975</b>	<b>0.16</b>	<b>0.2716</b>	<b>0.9978</b>	<b>0.0003</b>	<b>0.9978</b>	<b>0.9978</b>	<b>0.9977</b>	<b>0.9975</b>	<b>0.9997</b>

TABLE 9: Statistical performance measures along with their respective means (styled in bold) achieved by the proposed technique on handwritten *Telugu* numerals.

Class	Statistical performance measures									
	Kappa statistics	MAE	RMSE	TPR	FPR	Precision	Recall	<i>F</i> -measure	MCC	AUC
"0"				0.980	0.001	0.990	0.980	0.985	0.983	0.988
"1"				0.970	0.002	0.980	0.970	0.975	0.972	0.991
"2"				1.000	0.002	0.980	1.000	0.990	0.989	1.000
"3"				0.990	0.000	1.000	0.990	0.995	0.994	0.999
"4"	0.9867	0.0051	0.0449	0.990	0.002	0.980	0.990	0.985	0.983	0.998
"5"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"6"				0.990	0.003	0.971	0.990	0.980	0.978	0.995
"7"				0.990	0.002	0.980	0.990	0.985	0.983	1.000
"8"				1.000	0.000	1.000	1.000	1.000	1.000	1.000
"9"				0.970	0.000	1.000	0.970	0.985	0.983	0.994
Mean	<b>0.9867</b>	<b>0.0051</b>	<b>0.0449</b>	<b>0.988</b>	<b>0.0012</b>	<b>0.9881</b>	<b>0.988</b>	<b>0.988</b>	<b>0.9865</b>	<b>0.9965</b>

on limited dataset. (2) Training and testing times are not mentioned in most of the works. (3) Most of the works have been done for *Roman* because of the availability of larger dataset like MNIST. (4) Recognition systems for *Indic* scripts are mainly focused on single script. (5) Limitation to some feature extraction methods also exist; that is, they are local to a particular script/language rather having a global scope. In this work, we have verified the effectiveness of a moment based approach to handwritten digit recognition problem that includes geometric moment, moment invariant, affine moment invariant, Legendre moment, Zernike moment, and complex moment. The present scheme has been tested for five different popular scripts, namely, *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu*. These methods have been evaluated on the *CMATER* and MNIST databases using multiple classifiers. Finally, MLP classifier is found to produce the highest recognition accuracies of 99.3%, 99.5%, 98.92%, 99.77%, and 98.8% on *Indo-Arabic*, *Bangla*, *Devanagari*, *Roman*, and *Telugu* scripts, respectively. The results have demonstrated that the application of moment based approach leads to a higher accuracy compared to its counterparts. Among the most important ones, an advantage of this feature extraction algorithm is that it is less computationally

expensive where the most of the published works need more computation time. These features are also very simple to implement compared to other methods. It is obvious that, to improve the performance of proposed system further, we need to investigate more the sources of errors. Potential moment features other than the presented ones may also exist.

To further improve the performance, possible future works are as follows: (1) although the moment based features perform superbly on the whole, complementary features like concavity analysis may help in discriminating confusing numerals. For example, *Indo-Arabic* numerals "2" and "3" can better be separated by considering the original size before normalization. (2) For classifier design, it is better to select model parameters (classifier structures) by cross validation rather than empirically as done in our experiments. (3) Combining multiple classifiers can improve the recognition accuracy.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments


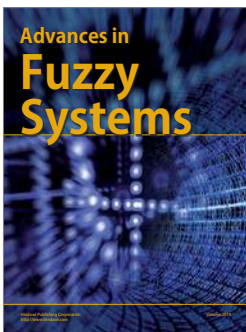
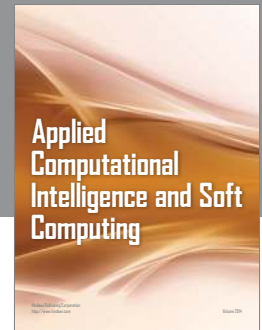
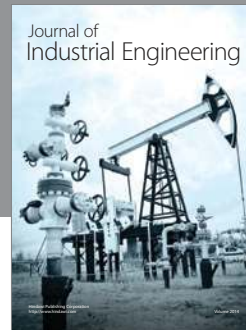
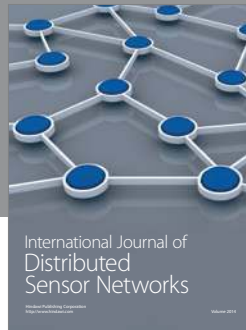
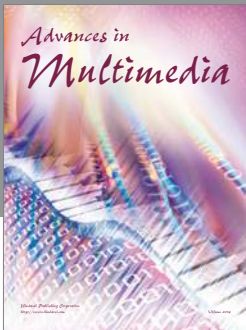
The authors are thankful to the Center for Microprocessor Application for Training Education and Research (CMATER) and Project on Storage Retrieval and Understanding of Video for Multimedia (SRUVM) of Computer Science and Engineering Department, Jadavpur University, for providing infrastructure facilities during progress of the work. The current work, reported here, has been partially funded by University with Potential for Excellence (UPE), Phase-II, UGC, Government of India.

## References

- [1] P. K. Singh, R. Sarkar, and M. Nasipuri, "Offline Script Identification from multilingual Indic-script documents: a state-of-the-art," *Computer Science Review*, vol. 15-16, pp. 1–28, 2015.
- [2] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, no. 10, pp. 2271–2285, 2003.
- [3] D. Gorgevik and D. Cakmakov, "Handwritten digit recognition by combining SVM classifiers," in *Proceedings of the International Conference on Computer as a Tool (EUROCON '05)*, vol. 2, pp. 1393–1396, Belgrade, Serbia, November 2005.
- [4] M. D. Garris, J. L. Blue, and G. T. Candela, *NIST Form-Based Handprint Recognition System*, NIST, 1997.
- [5] X. Chen, X. Liu, and Y. Jia, "Learning handwritten digit recognition by the max-min posterior pseudo-probabilities method," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 342–346, Parana, Brazil, September 2007.
- [6] K. Labusch, E. Barth, and T. Martinetz, "Simple method for high-performance digit recognition based on sparse coding," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1985–1989, 2008.
- [7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [8] Y. Wen, Y. Lu, and P. Shi, "Handwritten Bangla numeral recognition system and its application to postal automation," *Pattern Recognition*, vol. 40, no. 1, pp. 99–107, 2007.
- [9] V. Mane and L. Ragha, "Handwritten character recognition using elastic matching and PCA," in *Proceedings of the International Conference on Advances in Computing, Communication and Control*, pp. 410–415, ACM, Mumbai, India, January 2009.
- [10] R. M. O. Cruz, G. D. C. Cavalcanti, and T. I. Ren, "Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble," in *Proceedings of the 17th International Conference on Systems, Signals and Image Processing*, pp. 215–218, Rio de Janeiro, Brazil, June 2010.
- [11] B. V. Dhandra, R. G. Benne, and M. Hangarge, "Kannada, telugu and devanagari handwritten numeral recognition with probabilistic neural network: a script independent approach," *International Journal of Computer Applications*, vol. 26, no. 9, pp. 11–16, 2011.
- [12] J. Yang, J. Wang, and T. Huang, "Learning the sparse representation for classification," in *Proceedings of the 12th IEEE International Conference on Multimedia and Expo (ICME '11)*, pp. 1–6, IEEE, Barcelona, Spain, July 2011.
- [13] A. Giménez, J. Andrés-Ferrer, A. Juan, and N. Serrano, "Discriminative bernoulli mixture models for handwritten digit recognition," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11)*, pp. 558–562, IEEE, Beijing, China, September 2011.
- [14] Y. Al-Ohali, M. Cheriet, and C. Suen, "Databases for recognition of handwritten Arabic cheques," *Pattern Recognition*, vol. 36, no. 1, pp. 111–121, 2003.
- [15] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application," *Applied Soft Computing*, vol. 12, no. 5, pp. 1592–1606, 2012.
- [16] M. S. Akhtar and H. A. Qureshi, "Handwritten digit recognition through wavelet decomposition and wavelet packet decomposition," in *Proceedings of the 8th International Conference on Digital Information Management (ICDIM '13)*, pp. 143–148, IEEE, Islamabad, Pakistan, September 2013.
- [17] Z. Dan and C. Xu, "The recognition of handwritten digits based on BP neural network and the implementation on Android," in *Proceedings of the 3rd International Conference on Intelligent System Design and Engineering Applications (ISDEA '13)*, pp. 1498–1501, IEEE, Hong Kong, January 2013.
- [18] U. R. Babu, Y. Venkateswarlu, and A. K. Chinthla, "Handwritten digit recognition using K-nearest neighbour classifier," in *Proceedings of the World Congress on Computing and Communication Technologies (WCCCT '14)*, pp. 60–65, Trichirappalli, India, March 2014.
- [19] J. H. AlKhateeb and M. Alseid, "DBN—based learning for Arabic handwritten digit recognition using DCT features," in *Proceedings of the 6th International Conference on Computer Science and Information Technology (CSIT '14)*, pp. 222–226, Amman, Jordan, March 2014.
- [20] S. Abdleazeem and E. El-Sherif, "Arabic handwritten digit recognition," *International Journal on Document Analysis and Recognition*, vol. 11, no. 3, pp. 127–141, 2008.
- [21] R. Ebrahimzadeh and M. Jampour, "Efficient handwritten digit recognition based on Histogram of oriented gradients and SVM," *International Journal of Computer Applications*, vol. 104, no. 9, pp. 10–13, 2014.
- [22] A. M. Gil, C. F. F. C. Filho, and M. G. F. Costa, "Handwritten digit recognition using SVM binary classifiers and unbalanced decision trees," in *Image Analysis and Recognition*, vol. 8814 of *Lecture Notes in Computer Science*, pp. 246–255, Springer, Basel, Switzerland, 2014.
- [23] B. El Qacimy, M. A. Kerroum, and A. Hammouch, "Feature extraction based on DCT for handwritten digit recognition," *International Journal of Computer Science Issues*, vol. 11, no. 6(2), pp. 27–33, 2014.
- [24] S. AL-Mansoori, "Intelligent handwritten digit recognition using artificial neural network," *International Journal of Engineering Research and Applications*, vol. 5, no. 5, pp. 46–51, 2015.
- [25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, vol. I, Prentice-Hall, New Delhi, India, 1992.
- [26] F. Hausdorff, "Summationsmethoden und Momentfolgen. I," *Mathematische Zeitschrift*, vol. 9, no. 1-2, pp. 74–109, 1921.
- [27] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [28] M. Petrou and A. Kadyrov, "Affine invariant features from the trace transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 30–44, 2004.



- [29] P.-T. Yap and R. Paramesran, "An efficient method for the computation of Legendre moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1996–2002, 2005.
- [30] S. X. Liao and M. Pawlak, "On image analysis by moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 254–266, 1996.
- [31] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.
- [32] Y. S. Abu-Mostafa and D. Psaltis, "Recognitive aspects of moment invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 698–706, 1984.
- [33] Y. S. Abu-Mostafa and D. Psaltis, "Image normalization by complex moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 1, pp. 46–55, 1985.
- [34] August 2015, [http://en.wikipedia.org/wiki/Languages\\_of\\_India](http://en.wikipedia.org/wiki/Languages_of_India).
- [35] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI '95)*, pp. 338–345, San Mateo, Calif, USA, August 1995.
- [36] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.
- [37] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [38] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [39] S. le Cessie and J. C. van Houwelingen, "Ridge estimators in logistic regression," *Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992.
- [40] P. K. Singh, R. Sarkar, N. Das, S. Basu, and M. Nasipuri, "Statistical comparison of classifiers for script identification from multi-script handwritten documents," *International Journal of Applied Pattern Recognition*, vol. 1, no. 2, pp. 152–172, 2014.
- [41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

