A Case Study on Handwritten *Indic* Script Classification: Benchmarking of the Results at Page, Block, Text-line, and Word Levels

PAWAN KUMAR SINGH, Department of Information Technology, Jadavpur University, Kolkata, West Bengal, INDIA
RAM SARKAR, Department of Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, INDIA
AJITH ABRAHAM, Machine Intelligence Research (MIR) Labs, Scientific Network for Innovation and Research Excellence, Auburn, Washington, USA and Center for Artificial Intelligence, Innopolis University, Russia
MITA NASIPURI, Department of Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, INDIA

Handwritten script classification is still considered as a challenging research problem in the domain of document image analysis. Although some research attempts have been made by the researchers for solving the challenging issues, a comprehensive solution is yet to be achieved. The case study, undertaken here, analyzes the performances of various state-of-the art handwritten script classification methods for Indian scripts where features, needed for the script classification task, are extracted from the script images at four different granularity levels, i.e., page, block, text line, or word. The results of handwritten script classification at each level have been obtained and compared using eight different feature sets and six different state-of-the-art classifiers. Based on the classification results, an ideal level for performing the handwritten script classification task is suggested among these four classification levels. The results have also been improved by using two feature dimensionality reduction methods. All these experiments are done on two different handwritten *Indic* script databases, of which one is an in-house developed dataset and the other one is a freely available dataset. Finally, some future research directions that may be undertaken by the researchers as an application of the handwritten *Indic* script classification problem are also highlighted. The work presented here provides a basic foundation for the construction of a comprehensive handwritten script classification method for official Indian scripts.

Q1 CCS Concepts: → Computing methodologies → Artificial intelligence → Computer vision → Computer vision representations;

Additional Key Words and Phrases: Handwritten script classification, *Indic* scripts, case study, structure based features, visual appearance-based features

Authors' addresses: P. K. Singh, Department of Information Technology, Jadavpur University, Kolkata-700106, West Bengal, India; email: pawansingh.ju@gmail.com; R. Sarkar and M. Nasipuri, Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, West Bengal, India; emails: {raamsarkar, mitanasipuri}@gmail.com; A. Abraham, Machine Intelligence Research (MIR) Labs, Scientific Network for Innovation and Research Excellence, Auburn, Washington 98071, USA and Center for Artificial Intelligence, Innopolis University, Russia; email: ajith.abraham @ieee.org.

2375-4699/2021/10-ART32 $15.00
https://doi.org/10.1145/3476102

ACM Reference format:
Pawan Kumar Singh, Ram Sarkar, Ajith Abraham, and Mita Nasipuri. 2021. A Case Study on Handwritten *Indic* Script Classification: Benchmarking of the Results at Page, Block, Text-line, and Word Levels. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 21, 2, Article 32 (October 2021), 36 pages.
https://doi.org/10.1145/3476102

1 INTRODUCTION

Documents containing textual information of text printed/written in any one of different scripts prevalent in a multilingual country are known as multi-script documents. To handle such documents, even manually, grouping of documents on the basis of scripts used therein is an essential prerequisite. Script classification is still considered as a challenging research problem for any multi-script or multilingual environment [1]. However, in comparison with the huge volume of work exist in the domain of **Optical Character Recognition (OCR)** and analysis of documents, the amount of work on classification of scripts for Indian script documents is comparatively less [2]. One of the reasons for this is that research in the field of OCR domain has been generally concentrated at resolving problems within the purview of the country where the study has been organized. Since many of the nations around the world use only one script, study related to developing multi-script OCR systems in these nations has not got much attention [3]. However, identification of diverse scripts using a single OCR system is practically impossible. This is the reason that attributes essential for OCRing depend on the fundamental characteristics, nature, and flair of writing that usually vary for different scripts. Handwritten script classification plays a crucial role in a multi-script environment where the script of the handwritten document pages that contain text-blocks or text-lines or simply some isolated words is recognized [4].

Classification of handwritten scripts poses huge challenges than that of printed scripts [5–8]. The reasons for this are as follows: (a) The texture (feel and appearance) of printed scripts is much more distinctive than the handwritten ones. (b) The pixel intensity values for unrelated scripts are almost identical in case of handwritten script identification. (c) The straightness of vertical and horizontal strokes and the symmetry with respect to boundary points in the character image of scripts are much more unique in nature for printed scripts in comparison with handwritten scripts. Additionally, the stroke width is constant for a particular font type and size in case of printed scripts. (d) The character shape and size of the printed scripts are unique in each font type, whereas for handwritten scripts, the shape and size of the texts depend on the writing styles of each individual. (e) The *intra-word* and *inter-word* spacings for printed scripts are almost constant. (f) The pixel density and smoothness of the characters (particularly at curved shapes) for printed scripts are much more uniform. This, in turn, makes the feature values much more uniform for printed scripts rather in contrast to handwritten scripts. Apart from the above-mentioned reasons, some typical problems such as skew, noise, word crumbling caused by poor contrast, ruling lines, and so on, are more prominent in handwritten documents than in printed ones.

Automatic script classification in a multi-script situation is a difficult and thought-provoking research topic as observed from research works done over the past 20 years [9]. Researchers have come up with few feature-based approaches to solve the problem especially for *Indic* scripts considering the input information either as a document page, or a text block, or a text-line or simply a word [10, 11]. However, the difficulty lies in the fact that there is hardly any work found in the literature that is performed to apply the same feature sets at different levels to decide the optimal level of input to be considered. This is the primary point of motivation for designing a case study for analyzing the performances of previously proposed handwritten script classification works for handwritten *Indic* scripts at the said four different levels. This would, in turn, help to study and

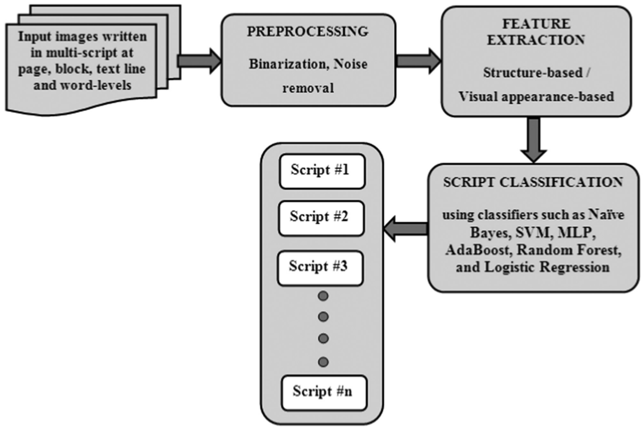


Fig. 1. Schematic diagram representing the typical handwritten script classification system.

examine (or compare) different feature sets using different classifiers found in the literature till date. Figure 1 shows the schematic diagram of the case study for designing any handwritten script classification method.

The content of this case study is ordered in the following manner: A short description regarding different types of languages and scripts used in India is presented in Section 2. In Section 3, a brief literature survey is carried out to revisit the preliminary works performed for classification of handwritten *Indic* scripts. The motivation behind the development of a case study for handwritten *Indic* script classification in a multi-script environment is mentioned in Section 4. It also describes some previous previously proposed feature descriptors applied on handwritten script images at four different levels. Section 5 presents the preparation of page, block, text-line, and word level datasets for performing the current experimentation as well as detailed analysis of the results of script classification outcomes. The scope of future research directions related to handwritten *Indic* script classification is mentioned in Section 6. Last, some conclusive remarks related to this case study is also reported in Section 7.

2 BRIEF OVERVIEW OF INDIAN SCRIPTS

The term "script" can be described as the basic graphical way of the writing which expresses languages in a written form [12]. In a multilingual country like India, 23 languages are recognized constitutionally, which consist of *English* and 22 Indian languages such as *Malayalam, Gujarati, Marathi, Konkani, Assamese, Oriya, Urdu, Bangla, Telugu, Kashmiri, Sindhi, Nepali, Tamil, Punjabi, Sanskrit, Kannada, Bodo, Manipuri, Dogari, Hindi, Maithili,* and *Santhali. English* is used as a provisional official language of Indian Union and used by nearly 125 million people of India. Moreover, the government of India has given the distinction of classical language to *Odia, Tamil, Malayalam, Kannada, Sanskrit,* and *Telugu.* Table 1 illustrates the 22 constitutionally recognized languages used in the Indian sub-continent listed in the "*Eighth schedule defined in May 2007*" [13], including the number of native speakers as well as the regions where these languages are widely used. Figure 2 illustrates the inscription of different Indian languages on a 2000 Indian currency note.

Languages used around the globe are built of various scripts, though a single script can be shared by different languages. For instance, the *Devanagari* script is used to write many Indian languages such as *Konkani, Sanskrit, Nepali, Hindi*, and so on. *Hindi,* among *Indic* languages, is spoken by nearly 500 million people. Currently, 12 official Indian scripts are prevalent in India, which are as follows: *Tamil, Odia, Bangla, Urdu, Gurumukhi, Devanagari, Malayalam, Kannada, Telugu, Roman,*

Table 1. Basic Information Related to 22 Indian Languages (and its Major Scripts) in Terms of Number of Native Speakers and the States in Which These Are Widely Used

Sl. No.	**Language**	**Number of native speakers (Millions)**	**Major script used**	**Place(s)/Community**
1.	*Assamese/Asomiya*	16.8	*Bangla*	Assam
2.	*Bengali/Bangla*	181	*Bangla*	Andaman & Nicobar Islands, Tripura, West Bengal
3.	*Bodo*	0.5	*Devanagari*	Assam
4.	*Dogri*	3.8	*Devanagari*	Jammu region of Jammu and Kashmir and Himachal Pradesh, northern Punjab
5.	*Gujarati*	46.5	*Gujarati*	Dadra and Nagar Haveli, Daman and Diu, Gujarat
6.	*Hindi*	182	*Devanagari*	Andaman and Nicobar Islands, Maharashtra Arunachal Pradesh, Bihar, Chandigarh, Chhattisgarh, the national capital territory of Delhi, Haryana, Himachal Pradesh, Jharkhand, Madhya Pradesh, Rajasthan, Uttar Pradesh and Uttarakhand.
7.	*Kannada*	3.63	*Kannada*	Karnataka. Listed as a Classical Language of India in 2008.
8.	*Kashmiri*	5.6	*Urdu*	Jammu and Kashmir
9.	*Konkani*	7.6	*Devanagari*	Goa, Karnataka, Maharashtra
10.	*Maithili*	34.7	*Devanagari*	Bihar
11.	*Malayalam*	35.9	*Malayalam*	Kerala, Andaman and Nicobar Islands, Lakshadweep. Listed as a Classical Language of India in 2013.
12.	*Manipuri/Meithei*	13.7	*Manipuri*	Manipur
13.	*Marathi*	68.1	*Devanagari*	Dadra & Nagar Haveli, Daman and Diu, Goa, Maharashtra
14.	*Nepali*	13.9	*Devanagari*	Sikkim, West Bengal
15.	*Odia*	31.7	*Odia*	Odisha. Listed as a Classical Language of India in 2014.
16.	*Punjabi*	1.05	*Gurumukhi*	Chandigarh, Delhi, Haryana, Punjab
17.	*Sanskrit*	0.03	*Devanagari*	Listed as a Classical Language of India in 2005.
18.	*Santhali*	6.2	*Roman*	Santhal tribals of the Chota Nagpur Plateau (comprising the states of Bihar, Chhattisgarh, Jharkhand, Orissa)
19.	*Sindhi*	21.4	*Devanagari*	Sindhi community
20.	*Tamil*	65.7	*Tamil*	Tamil Nadu, Andaman & Nicobar Islands, Kerala, Pondicherry. Listed as a Classical Language of India in 2004.
21.	*Telugu*	69.8	*Telugu*	Andaman & Nicobar Islands, Andhra Pradesh. Listed as a Classical Language of India in 2008.
22.	*Urdu*	60.6	*Urdu*	Andhra Pradesh, Delhi, Jammu and Kashmir, Uttar Pradesh, Tamil Nadu

Fig. 2. Different languages (highlighted in red color) carved on a 2000 INR currency note illustrating the diversity of India in terms of languages used [13].

Gujarati and *Manipuri*. Among these, the 10 scripts excluding *Roman* and *Urdu* are known as *Indic* scripts. They are generally referred to as "script composition grammar" (English!), as they do not have any sub-component in all remaining groups of scripts across the globe [14]. *Indic* scripts are generally written with the help of syllables and can be optically deconstructed into three tiers where the constituent symbol of each tier plays a pre-specified role for the interpretation of the syllable.

3 LITERATURE STUDY

A comprehensive literature study describing the feature extraction and classification methodologies for printed and handwritten script recognition of *Indic* scripts is reported by Singh et al. in 2015 [15] and Sahare et al. in 2017 [16]. The problem of script classification from handwritten texts is quite popular in the domain of document image processing but most of the works have been done for the same for non-*Indic* scripts rather than for *Indic* scripts (as described by Ubul et al. in 2017 [17]). The basis of script recognition is based on the collection of characters of different scripts that vary by a significant amount. Also, the different spatial orientation, alignment, and the visual features of the characters of different *Indic* scripts give rise to the unique properties to the scripts. These make it possible to identify the scripts in which a particular text is written. So, the main task researchers generally follow is to develop a complete model that can acquire these different attributes for a script image, and then classify accordingly. Based on these methods and the types of features used, we can classify these methods into 2 major types – (a) Structure-based approaches and (b) Visual appearance-based approaches. Scripts, in general, differ from each other in the terms of writing style, stroke structure, stroke connection, orientation associated with the character sets [18].

Structure-based approaches are those techniques, where the connected components are obtained from the script document images, and then their components are analyzed using different shape and structure based features. A number of research papers have been published by applying Structure-based approaches. Using Structure-based approaches, handwritten script classification is done depending on the attributes (or features) obtained from text words [19–21]. These techniques are usually applied to obtain attributes (features) after applying the **CCL (Connected**

Component Labeling) algorithm for defining the textual part of script. In the work described by Singh et al. [19], a set of 39 structural features is obtained using convex hull of script words inscribed in *Devanagari* and *Roman* scripts. The recognition procedure is done by using the **Multi-Layer Perceptron (MLP)** classifier. Singh et al. [20] designed a handwritten script classification method for seven Indian scripts *viz., Roman, Telugu, Devanagari, Bangla, Malayalam, Gurumukhi*, and *Odia* at word level. An 82-element feature vector is constructed by combining the elliptical and polygon approximation-based approaches. In recent times, Obaidullah et al. [21] introduced a unique methodology to classify Indian scripts using the "Matra" as the discriminating factor. This method used the concept of an improved fractal geometry analysis, and Random forest classifier is used as the classification algorithm for improving the performance of script classification from the images of handwritten script documents. Obaidullah et al., in Reference [22], reported a multi-level script identification scheme to pick the optimal portion of a document image on which the handwritten script identification method may be performed successfully. Script-independent and script-dependent features are identified in their work. A qualitative measure of these two feature sets is then computed at individual level for grouping different scripts at different levels. But the focus of this work mainly depends upon Structural-based features (of size 56 only), which may not be enough for solving this challenging problem. Additionally, the script classification task is carried out using only two classifiers, namely, MLP and Random forest. The same authors [23] introduced and compared an **extreme learning machine (ELM)** classification methodology having five different activation functions for text-line level handwritten *Indic* script classification. The testing of the scripts were done on bi-script, tri-script, and multi-script levels, and the highest recognition accuracy was achieved using sigmoidal function.

In visual appearance-based approaches, the texture-based features are used that are obtained from the textual part of a script document, i.e., the attributes (or features) are derived from the way the characters are grouped into set of components and eventually into words (What is the meaning of this?). Several research works use the texture features such as **Histogram of Oriented Gradients (HOG)** [24], **Gray Level Co-occurrence Matrix (GLCM)** [25, 26], a combination of **Discrete Wavelet Transform (DWT)** and **Radon Transform (RT)** [27], **Modified log-Gabor (MLG)** Transform [28], combination of **Neighborhood Gray Tone Difference Matrix (NGTDM)** and **Gray Level Run Length Matrix (GLRLM)** [29], **Distance Hough Transform (DHT)** algorithm [30], and so on. Hangarge et al. [31] presented a script classification method depending on a directional **Discrete Cosine Transform (DCT)** for *Telugu, Kannada, Tamil, Roman, Malayalam*, and *Devanagari* scripts. Pardeshi et al. [32] introduced a multi-dimensional feature set by combining RT, DWT, DCT, Statistical filter approaches (??) for the classification of 11 different scripts at word level. Obaidullah et al. [33] proposed a block level script classification approach for the document images written in *Odia, Roman, Malayalam, Devanagari, Urdu,* and *Bangla* scripts. At first, a feature set comprising 34 elements was formed by combining DT, **Fast Fourier Transform (FFT)**, DCT, and RT. Then, a feature vector of size 20 was chosen using Greedy Attribute Selection scheme and the MLP classifier was used to classify the input scripts. The technique was tested on unconstrained handwritten dataset comprising 600 script text blocks of pre-defined size 512×512 pixels. Average bi-script, tri-script, and tetra-script recognition accuracies were found to be 95.33%, 88.89%, and 87.18%, respectively. Obaidullah et al. [34] proposed a multi-script handwritten word level image dataset for 11 different scripts. The dataset contains approximately 300 text words per script that was not made freely available. Result on the dataset was benchmarked using three different feature sets (extracted from spatial energy, wavelet energy and RT) and three different classifiers such as MLP, **Fuzzy Unordered Rule Induction Algorithm (FURIA)** and Random forest. An average accuracy of about 98.60% was reported using MLP classifier.

Vijayalaxmi et al. [35] described a block level handwritten script recognition methodology for *Tamil, Kannada, Devanagari, Roman, Malayalam,* and *Telugu* scripts. The classification was done with the help of features extracted from GLCM and multi-resolutionality. A set of 23 spatial features were obtained from 2D-DWT, whereas a 20-dimensional feature set based on texture features was obtained from the GLCM feature descriptor. The proposed technique was evaluated on a self-prepared database comprising 600 script text blocks, and the experiments were conducted at both tri-script and bi-script levels. Classification of the input scripts was performed with the help of MLP, **Support Vector Machine (SVM)**, and **Linear Discriminant Analysis (LDA)** classifiers. Finally, the best script recognition accuracies of 96.43% and 93.98% using SVM classifier were obtained at bi-script and tri-script levels, respectively.

3.1 Motivation

From the literature study, we have witnessed that in the context of handwritten Indian script classification, mostly researchers have implemented their methods considering a certain number of Indian scripts at a particular level. No work has been found to date accomplishing all the features on a common dataset and applying them at all the four levels of classification. This type of research work is needed to decide the optimal level of handwritten script classification. It is known that multi-script environment is a common aspect in Indian sub-continent. Therefore, methods developed considering few Indian scripts can be a major shortcoming in the practical scenario. Hence, in Indian context, there lies a pressing need for performing a case study that, in turn, would help the researchers to understand the problem domain clearly and help them to develop a comprehensive handwritten script classification system considering all the *Indic* scripts at all the four levels.

4 HANDWRITTEN *INDIC* SCRIPT CLASSIFICATION: A CASE STUDY

It is evident from the literature survey that the research done on script classification for handwritten documents only considers the input given at a specific level that is, page level [22, 26, 28], block level [22, 33, 35], text-line level [22, 23, 25, 29], or word level [18–22, 24, 27, 30–32]. The overall thought is that the script classification results obtained at page level is much better as compared to block, text-line, and word levels. The reason lies in the fact that the textual contents present in page level are more widespread than at block, text-line, and word levels. Alternatively, the script classification performance at block level may be lower because of the existence of fragmented and inadequate textual contents as a result of segmentation. The script data at both text-line and word levels are also found to be inconsistent sometimes, as the amount of connected components might be insignificant here for performing the script classification task. However, the effectiveness of text-line and word segmentation algorithms (from the handwritten script documents) merely determine the overall performance of text-line level and word level handwritten script classification methods, respectively. Since the information present at word level is very less as compared to the other three levels, the script classification results achieved at word level is pretty less as compared to the other three levels of classification. To verify the aforementioned facts, the present case study is aimed at conducting the experiments on Indian handwritten script databases prepared at four different levels. Besides, script classification from handwritten script documents at page, block, text-line, or word levels is performed and their individual outcomes are compared and validated. To be more precise, a specific script document is considered at all four distinct levels and the individual classification results are compared when the script information varies from one level to another. The analysis for justifying the suitability of the features (considered in the present case study) at different script levels is also made.

Hence, the contribution of this article is multi-fold:

(1) Preparing database written in 12 constitutionally enlisted Indian scripts at four different levels;
(2) Measuring the individual performance of handwritten script classification, i.e., classification of the input scripts from page, block, text-line, and word level images;
(3) Selecting an ideal level for handwritten script classification among the said four different levels;
(4) Evaluating performances of different feature sets on four different script levels, which has been noted and statistically validated using six state-of-the-art classifiers that include Naïve Bayes, SVM, MLP, AdaBoost, Random Forest, and Logistic Regression;
(5) Detailed performance analysis of the best case of handwritten script classification results;
(6) Improving the overall script classification performance by using feature dimensionality reduction methods;
(7) Evaluating performance on two different handwritten *Indic* script databases of which one is an in-house developed dataset and the other one is a freely available dataset; and
(8) Scope of future research directions that may be carried out as an application to handwritten *Indic* as well as other script classification.

4.1 Features

A total of eight feature sets are used in this case study, among them some are conventional and some are new. Even the conventional features that are used in this study are suitably customized to fit into the problem under consideration. Two structure-based feature sets are considered in this study, which are: (a) Convex-hull-based features and (b) Combination of Elliptical and polygonal approximation-based features. The remaining six feature sets are visual appearance-based features. They are as follows: (a) HOG, (b) GLCM, (c) Combination of DWT and RT, (d) MLG filter transform, (e) Combination of NGDTM and GLRLM, and (f) DHT algorithm.

4.1.1 Structure-based Features. Structure-based features attempt to capture some structural information of a script. However, presence of noise, skewness, inconsistent gap among *intra-words* (or *inter-words*) significantly affects the shape or structural information analysis procedures. However, one can apply these features for script classification at word level to avoid complexities of text-line or word segmentation. In this article, three structure-based features that are used for handwritten script recognition at word level are briefly explained in the subsequent subsections.

4.1.1.1 Convex-hull-based features. The concept of convex hull is extensively used in different pattern recognition problems, as it shows invariance towards scaling, translation, and rotation of the input images. It also shows efficient behavior for contour images (after filtering) affected with noise. Therefore, in the work described in Reference [19], 39 dimensional convex-hull-based features are estimated for the recognition of handwritten *Devanagari* and *Roman* script words. But, here, we deal with 12 Indian scripts instead of 2 scripts, so in the present work, a feature vector of size 145 is designed to incorporate more number of local features.

A parameter named d_{cp} is defined as the distances of data pixels measured row- and column-wise calculated from the boundary of the convex hull obtained from the upper, right-most, lower, and left-most boundaries of the script word image. Table 2 illustrates seven structural features estimated based on d_{cp}. In Table 2, the bays correspond to the region lying between the script word image and convex-hull perimeter. However, the lakes comprise the interior region covered within the script word image.

Since the feature values are computed from the upper, right-most, lower, and left-most boundaries of the script words, in total, 28 (i.e., 7×4) features are found. An additional feature is

Table 2. Description of Feature Vector Extracted from Convex Hull of an Image

Feature No.	Feature Description
F1	Maximum d_{cp}
F2	Total number of rows having $d_{cp} > 0$
F3	Average d_{cp}
F4	Total number of rows having $d_{cp} = 0$
F5	Mean row co-ordinate having $d_{cp} > 0$
F6	Number of visible bays
F7	Number of visible lakes

calculated along the convex-hull perimeter, which is defined as the count of the pixels located along the convex-hull perimeter with $d_{cp} = 0$. So, a feature vector of size 29 is computed from a handwritten script word constructed using convex hull. Now, to extract local information present in different script words, each script word is again divided into 4 sub-images depending on the centroid of the convex hull. For each of the sub-images, the convex hulls are then constructed and a feature vector comprising of 116 (i.e., 29×4) elements is again estimated using these sub-images of every script word. The overall feature vector is of size 145, which has 29 global features obtained from the entire script word image in addition to 116 local features computed using 4 sub-images of the same word image.

4.1.1.2 Elliptical-based Features. A set of attributes (features), proposed in Reference [20], depending on the hypothetically conceptualized regions of elliptical shape over the images of the script words has been constructed for identifying the scripts of the inputs at word level. These features are obtained from the boundary and the native areas of a script word therefore separating a specific script can be done at ease. For instance, in certain script of *Indic* alphabet (such as *Gurumukhi, Bangla*, and *Devanagari*, etc.) it is to note that several characters consist of straight line on the top part known as Shirorekha or Matra. With the help of these features, the Matra and non-Matra-based scripts (such as *Roman, Malayalam, Odia, Telugu*, etc.) can be differentiated as dissimilar scripts possess unlike pixel densities in certain explicit regions or zones. Two parameters, *namely*, P_r and P_c, are used to extract the features. P_r (*pixel ratio*) is the ratio of the count of boundary (contour) object (pixels) to the count of background pixels, and P_c (*pixel count*) is the count of boundary (contour) object (pixels). These two scenarios have been considered here.

In the first scenario, an ellipse has first been inscribed (taking into consideration the positioning of the ellipse) within this surrounding box containing minor and major axes equal to the height and the width of the surrounding box and the center of the corresponding surrounding box is same as the center of an ellipse. The word image is divided into 8 regions by this ellipse. Figure 3(a) shows this scenario for a sample handwritten *Tamil* word image. Estimation of 8 features of every handwritten script word image is done by considering P_r values from these 8 regions. Also, to calculate P_c, n lines are drawn parallel to both minor and major axes of the illustrative ellipse. The experimental value of n is taken to be 8 for this work. 4 other features are obtained using the standard deviation and mean of the values of $\mathrm{P_c}$ along minor/major axis. Every image of word bounded by the minimum bounding box is again divided into 4 identical rectangles and a demonstrative ellipse is fitted into each of these rectangles. With the help of the P_r values, computation of 32 feature values is done from the 32 regions in a like manner. It results in a feature vector of dimension 44 (that is, 8 + 2 + 32).

In the second scenario, the word image is circumscribed using a primary ellipse whose center is same as midpoint of its minimum bounding box. We consider the values of the minor and major

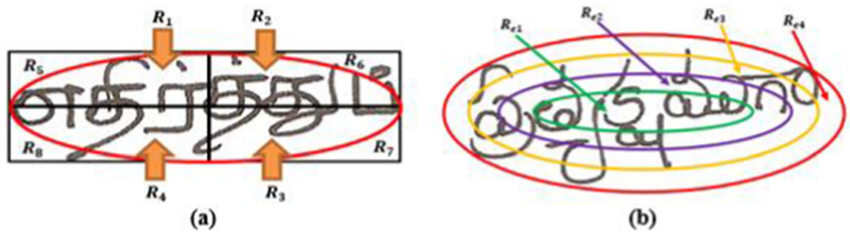


Fig. 3. Illustration of: (a) maximum inscribed ellipse fitted within a minimum bounding box of a handwritten *Tamil* word image that divides the word image into 8 regions, and (b) fitting four concentric ellipses over a sample *Telugu* word image. (recreated from Reference [20]).

axes of the ellipse. Three concentric ellipses are constructed within this ellipse after fitting it. These concentric ellipses have the identical center point as the primary ellipse (see Figure 3(b) for illustration). Estimation of 4 features values considering the P_c's and the P_r's is done from first, second, third, and fourth regions, respectively. The additional six features are considered as the differences between corresponding values of the P_r's and P_c's among first and second, second and third, third and fourth regions, respectively. This makes feature count as 14, (i.e., 4 + 4 + 6). Thus, a feature vector of size 58 (44 + 14) is considered from the elliptical features.

4.1.1.3 Polygonal Approximation. The main aim of polygonal approximation is to arrest the crux of the shape in a specified periphery (boundary) with the help of fewer number of parts (segments). There are two methods used to realize Polygonal approximation [36]. They are: (a) Distance Threshold method and (b) Fit and Split method.

In the first method, the boundary points of the script images coincide as long as the **Least Square Error (LSE)** fits to the points fused up to cross a predetermined threshold. With the occurrence of this situation, the line parameters are kept and the value "0" is assigned to error. The process is followed again and again by joining new boundary points till the threshold lies below the error. Last, the vertices of the polygon are formed by intersecting the adjacent line segments. The value of μ (known as threshold) is considered as 3. Finally, a 12-bin histogram is generated with the help of the maximum distance calculated between the vertices lying on the input curve from its line segments. This 12-bin histogram is considered as the feature vector.

In the second method, a boundary segment is subdivided consecutively into two portions until a definite criterion satisfies. First, we find two points on the boundary that are farthest away and draw a line between them. For each boundary segment, we find a point on the boundary that has a maximum perpendicular distance to its corresponding line. Finally, we draw lines joining the boundary point and the two end points, respectively, of the corresponding splitting line to form the approximate polygon. The entire procedure is repeated until the perpendicular distance is less than a threshold. This method possesses the benefit of searching noticeable points of inflection. Feature values have been calculated with the help of an identical rule used in Distance Threshold technique to calculate a set of 12 feature values. Last, an 82-dimensional feature vector is constructed by combining both the polygonal approximation and elliptical-based features [20].

4.1.2 Visual Appearance-based Features. The features extracted through texture analysis are known as visual appearance-based features. A texture can be defined as "*a repeated pattern of information or arrangement of the structure with regular intervals.*" "In a general sense, texture refers to surface characteristics and appearance of an object given by the size, shape, density, arrangement, proportion of its elementary parts [18]. Due to the significance of texture information, texture feature extraction plays a key role in various image processing applications like medical imaging, remote sensing and content-based image retrieval." In this case study, we have used six visual appearance-based features that are described briefly in the following subsections.

4.1.2.1 Histogram of Oriented Gradients (HOG). The concept of HOG descriptor is first invented by Dalal and Triggs in Reference [37] for object detection. They used these descriptors for the detection of pedestrians in still images. *Using HOG descriptors, the script images are first divided into smaller spatial sub-images named as "cells." The gradient direction of the pixels lying in the script images is calculated and a histogram is thus formed. The bins/channels of the designed histogram are uniformly set apart either between* 0^0 *to* 180^0 *(taking signed values of gradient) or* 0^0 *to* 360^0 *(taking unsigned values of gradient). Finally, this histogram is calculated for all the cells and the resultant feature vector is obtained by the combination of these histograms.* The advantage of using HOG feature descriptor is that it extracts the local information of the pixel orientation in every cell of the handwritten script images that, in turn, describes the appearance and structure in the present context. In the present case study, the number of cells is taken to be 10 and the histogram is constructed for 8 different bins. This produces an 80-dimensional feature vector using HOG descriptors [24].

4.1.2.2 Gray Level Co-occurrence Matrix (GLCM). GLCM [38] evaluates the characteristics of the script image based on second-order statistics that takes into consideration the association among pixels or groups of pixels. A set of 10 features such as *Entropy, Contrast, Cluster Prominence, Information Measure of Correlation, Covariance, Energy, Cluster Shade, Autocorrelation, Local homogeneity, and Inertia* using GLCM feature descriptor are estimated. The definitions related to these measurements are already detailed in References [25, 26]. Each of the above measurements is calculated for two different values of $d = \{1, 2\}$ and four different orientations $\theta = 0^0,\ 45^0, 90^0,\ \textit{and}\ 135^0$, which lead to 8 different features values. Hence, using GLCM feature descriptor, a feature vector of size 80 elements is obtained.

4.1.2.3 Combination of DWT and RT. The feature extraction methodology using a combination of DWT and RT (described in Reference [27]) is implemented to extract a feature set consisting of 48 elements for the recognition of handwritten scripts at all the four levels. "*In this method, the RGB word image is firstly converted into a gray scale image. The 2D discrete Haar wavelet transform [39] is then applied which transforms the script images into four different bands: LL, HL, LH, and HH. Here, LL denotes the approximation coefficient and HL, LH, and HH represent the detail coefficients. The upper left zone (LL) component is selected and binarized using Otsu's global thresholding approach [40]. The RT is then applied to the binarized LL component for 180 different orientations ranging from* 0^0 *to* 180^0 *. Three features such as standard deviation, kurtosis, and skewness are calculated along 12 different orientations (*$\theta = 0^0, 15^0, 30^0, 45^0, 60^0, 75^0, \ldots\ldots\ldots\ldots, 165^0$ *) of the RT matrix which makes the feature count as 36. Now, the binarized script images are inverted by considering foreground and background pixels as '0' and '1,' respectively. The RT is similarly applied to the inverted script images and standard deviation is calculated from this RT matrix along 12 different orientations. Finally, a feature vector of dimension 48 (36+12) is extracted from the combination of DWT and RT.*"

4.1.2.4 Modified log-Gabor Filter (MLG) Transform. The feature vector obtained using MLG (proposed in Reference [28]) has also been implemented for the recognition of handwritten *Indic* script images at page, block, text-line, and word levels. In the work described in Reference [28], a **Windowed Fourier Transform (WFT)** is taken into account for preserving the spatial information. The process of WFT involves two stages. In the first stage, the input image is multiplied with the window function. The FT is then applied to the previous step to get the resulting output at the final stage. In short, WFT is mainly a convolution of the low-pass filter with the input image. MLG transform uses a Gaussian function as the ideal concerted function in both spatial as well as frequency domain [41]. To get the filtered images as output, the inverse FT is finally implemented on the resulting script images. For obtaining the feature vector, two important measures such as energy and entropy features [39] are calculated from the MLG filter transformed images. Here, the number of scales (n_s) is chosen as 5 (that is, $n_s = 1, 2, 3, 4,$ and 5) and the number of orientations (n_0) is taken as 12 (that is, $0^0, 15^0, 30^0, 45^0, 60^0, 75^0, \ldots\ldots\ldots., 165^0$). Hence, this produces a feature set comprising 120 elements for a given input image containing handwritten text.

4.1.2.5 Combination of NGTDM and GLRLM. The difference between each pixel and the adjoining pixels in terms of gray-level intensity of the script images is calculated using the NGTDM feature descriptor [42]. Five different measurements, such as coarseness, contrast, busyness, complexity, and texture strength are computed using NGTDM, and the process is explained in Reference [29]. This is done for the quantitative assessment of characteristics of the resulting perceptual script textures. Two distances values such as $d = \{1, 2\}$ as well as two neighborhood sizes consisting of 3×3 and 5×5 pixels are considered for the feature extraction purpose. Therefore, using NGTDM feature descriptor, a 40-dimensional statistical textural feature vector is obtained for each input script image.

The concept of designing texture features using a run length matrix were first proposed by M. M. Galloway in Reference [43], A. Chu et al. in Reference [44] and B. R. Dasarathy et al. in Reference [45]. A set of 11 measurements such as "*Run Length Non-uniformity, Long Run Emphasis, Gray-level Non-uniformity, Short Run Emphasis, High Gray-level Run Emphasis, Low Gray-level Run Emphasis, Run Percentage, Long Run High Gray-level Emphasis, Short Run High Gray-level Emphasis Long Run Low Gray-level Emphasis, and Short Run Low Gray-level Emphasis*" are undertaken for the present case study. The four different values at different orientations as $\theta \in 0^0,\ 45^0, 90^0,\ and\ 135^0$ are considered. Thus, using GLRLM feature descriptor, a set of 44 statistical textural features are estimated. As a result, a feature vector of size 84 is obtained combining both NGTDM and GLRLM feature descriptors to classify 12 handwritten official *Indic* scripts.

4.1.2.6 Distance Hough Transform (DHT) Algorithm. A feature vector of size 72 (36 + 36) is considered using DHT algorithm (proposed in Reference [30]). DHT algorithm is a suitable combination of the **Hough transform (HT)** and **Distance transform (DT)**. *In this algorithm, the RGB script images are initially converted into gray scale images. These images are binarized using Otsu's global thresholding approach [40]. The image thinning is performed on the binarized script images. HT is then applied along 18 different orientations such as $\theta = -90^0, -80^0, -70^0, \ldots\ldots, 0^0, \ldots, 70^0, 80^0$), with ρ resolution taken as 1 pixel. The ρ -value corresponding to the maximum accumulator value is calculated along each orientation that produces a set of 36 features. The* ***Euclidean Distance transform (EDT)*** *is then applied to the script images. The transformed images are scaled by a factor of 8. Only the higher pixel values are taken into consideration and marked as '1,' whereas the remaining pixels in the images are marked as '0.' This is because the higher pixel values denote large distance that helps to analyze the shape and structure of the images written in different scripts. The thinning of the images is again performed to get the precise structure. A set of 36 features are then extracted in a similar fashion. This makes the total size of feature vector to 72 using DHT algorithm.*

Therefore, a total of eight feature descriptors (i.e., two structure-based features and six visual appearance-based features) have been applied for solving the problem of handwritten script recognition at four different levels. The summarization of these feature sets along with their dimension is also shown in Table 3. Moreover, one important point should be kept in mind: The two structure-based feature vectors are applicable for word level script identification only, whereas the six visual appearance-based feature vectors are applicable at all the four levels *viz.*, page, block, text-line, and word levels script classification.

Table 3. Summarization of the Feature Descriptors along with Their Dimension Used in the Present Case Study for Handwritten *Indic* Script Identification

Sl. No.	Feature Descriptor	Feature Dimension
Structure-based features		
1.	Convex-hull-based features [19]	145
2.	Elliptical based features and Polygonal Approximation [20]	82
Visual appearance-based features		
3.	HOG [24]	80
4.	GLCM [25, 26]	80
5.	DWT and RT [27]	48
6.	MLG Transform [28]	120
7.	NGTDM and GLRLM [29]	84
8.	DHT algorithm [30]	72

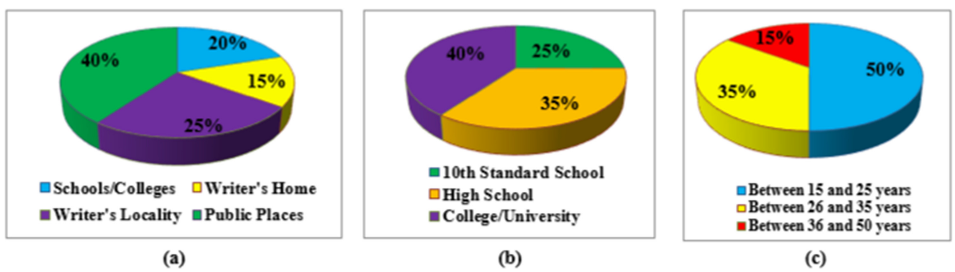
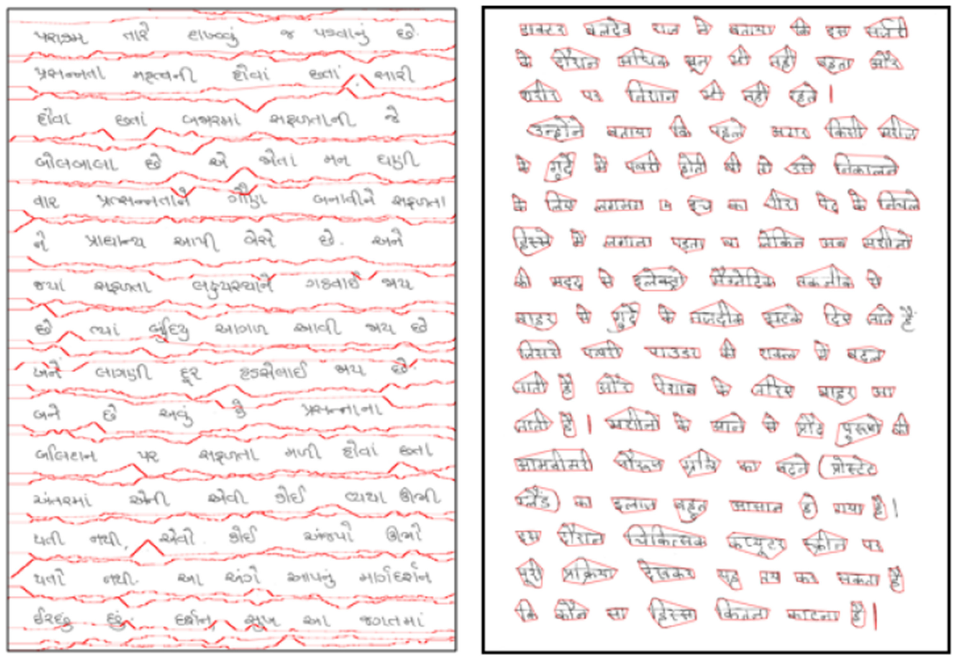


Fig. 4. Graphical representation of the distribution of the writers with respect to: (a) places of data collection, (b) educational level of writers, and (c) writers' ages.

5 BENCHMARK RESULTS FOR HANDWRITTEN *INDIC* SCRIPT CLASSIFICATION

This section presents the benchmark outcomes achieved for script classification from handwritten *Indic* script document pages of an in-house database at four different levels. The study also evaluates the performance of the above-mentioned eight feature sets using six sophisticated machine learning algorithms. For development of the in-house handwritten *Indic* script document image database, 180 diverse native writers in different age-groups and with different educational levels are selected. Volunteers of this data collection drive are requested to write text using a single script on A-4 size pages with a black or blue ink pen. No other restrictions are enforced concerning the content of the writing. This implies that writers are asked to write no matter what they want to write in their native script. Furthermore, the pages are compiled from different places (home, office, school, etc.) to include diverse groups of handwriting. Altogether, 95 males and 85 females participated in this data collection drive. The most important aspect of our developed database is the heterogeneity with respect to three important factors, namely, places of data collection, educational level, and age of the writers, as shown in Figures 4(a–c), respectively. The document pages are digitized at 300 dpi resolution and stored as gray tone images in 24-bitmap file format with the naming convention <Script>_###.bmp. Here, ### is a unique integer number for indexing the files and <Script> belongs to 12 different scripts. The pre-processing of the document images is done exactly in the same way as mentioned in Reference [27]. The developed handwritten *Indic* multi-script document image database will be made freely available to the research community.

Before conducting the experiments, a dataset of total 360 handwritten document images in 12 officially recognized *Indic* scripts, with exactly 30 document pages per script, is prepared from the developed database. Rectangular text block images of pre-defined size 256 × 256 pixels are



(a) (b)

Fig. 5. Output results of: (a) text-line segmentation (as described in Reference [46]) for handwritten *Gujarati* script document image and (b) word segmentation (as described in Reference [47]) for handwritten *Devanagari* script document image.

Table 4. Detailed Description of the Experimental Script Datasets

Dataset Name	Number of elements considered per script	Total number of elements
Page level	30 pages	360 pages
Block level	300 text blocks	3,600 text blocks
Text-line level	400 text-lines	4,800 text-lines
Word level	2500 words	30,000 words

extracted automatically from varied portions of the handwritten script images such that the handwritten script blocks either contain at least four or more word images with variable spaces or 50% of the script block region contains handwritten script. Next, the text-line and then the word images are also extracted from the input document pages using the techniques described in References [46] and [47], respectively. Sample results of text-line and word segmentation algorithms are illustrated in Figures 5(a–b), respectively. The datasets used in the experimentation are detailed in Table 4. It is observed from Table 4 that a total of 360 document pages, 3,600 text blocks, 4,800 text-lines, and 30,000 text words written in 12 official Indic scripts are considered for the script classification problem. Samples of text blocks, text-lines, and text words of the handwritten script dataset used in the experiment are shown in Figures (6–8), respectively. Two structural-based features (such as Convex-hull-based features [19] and combination of Elliptical and Polygonal Approximation features [20]) are extracted from word level datasets only, whereas six visual appearance-based features (such as HOG [24], GLCM [25, 26], combination of DWT and RT [27], MLG transform

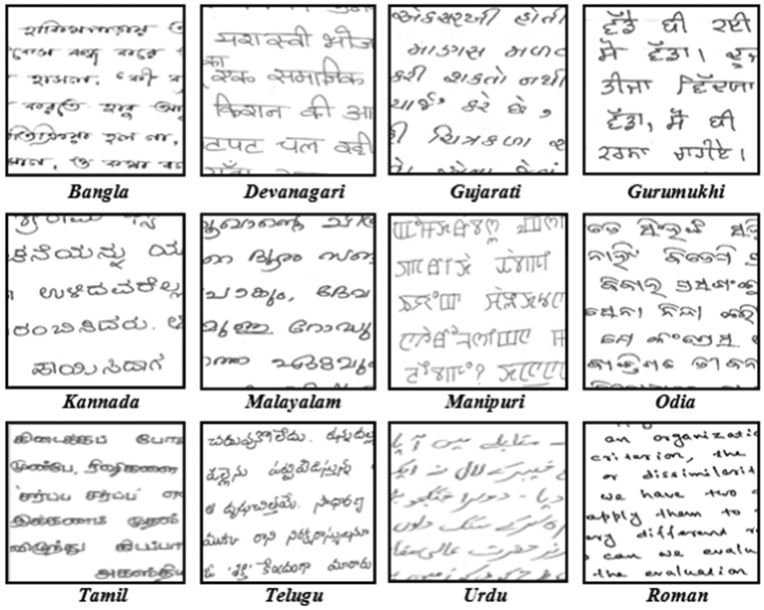


Fig. 6. Samples of handwritten *Indic* scripts prepared at block level.

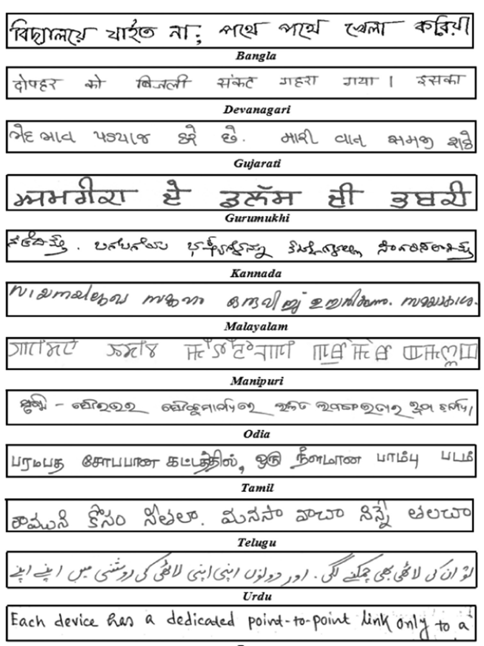


Fig. 7. Samples of handwritten *Indic* scripts prepared at text-line level.

Fig. 8. Samples of handwritten *Indic* scripts prepared at word level.

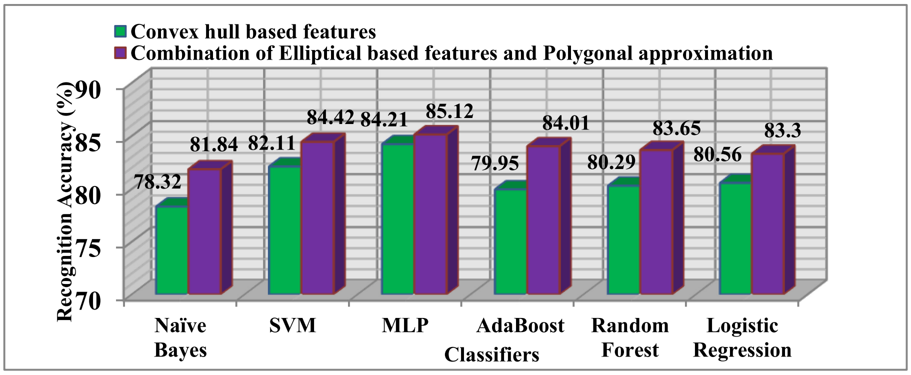


Fig. 9. Graphical comparison showing the recognition performances for two structural-based features, *namely*, convex-hull-based features [19] and combination of Elliptical and Polygonal approximation-based features [20] using six different classifiers.

[28], NGDTM and GLRLM [29], and DHT algorithm [30]) are extracted from each of the page, block, text-line, and word images of the datasets. These feature descriptors are briefly detailed in Section 4. The classification of the scripts is done using six popular state-of-the-art classifiers, namely, Naïve Bayes [48], RBF-SVM [49], MLP [50], AdaBoost [51], Random Forest [52], and Logistic Regression [53]. The whole experiment is conducted on the developed dataset using 3-fold cross-validation method. The performance of the mentioned feature sets in regard to their script classification ability is evaluated in terms of recognition accuracy as shown in Equation (1).

$$Recognition\ Accuracy = \frac{\#correctly\ classified\ components}{\#total\ number\ of\ components} \times 100\% \tag{1}$$

Additionally, we have measured the performance in terms of some standard parameters such as Kappa statistics, **Mean Absolute Error (MAE), Root Mean Square Error (RMSE), True Positive Rate (TPR), False Positive Rate (FPR)**, Precision, Recall, F-measure, and **Area Under ROC (AUC)**. The detailed definitions regarding these parameters can be found in one of our previous works described in Reference [27].

5.1 Benchmark Results of Handwritten Script Classification Using Structural-based Features

The script classification performances at word level using structural-based features of the aforementioned six classifiers are displayed in Figure 9. It is understood from Figure 8 that the MLP classifier realizes the best word level script recognition accuracies of 84.21% and 87.75% using convex-hull-based features and combination of Elliptical and Polygonal approximation-based features, respectively. The detailed script-wise performances at word level obtained by MLP classifier for each of these feature sets are shown in Tables (S1–S2), respectively, as separate supplementary material.

5.2 Benchmark Results of Handwritten Script Identification Using Visual Appearance-based Features

5.2.1 Using HOG Feature Descriptor. The script identification performance of the HOG feature descriptor [24] is studied at page, block, text-line, and word levels for each individual classifier. The results are illustrated in Table 5. It can be observed from Table 5 that the MLP classifier performs the best at all the four levels. The script recognition accuracies are found to be 90.11% at the page level,

Table 5. Individual Classifier Performance of HOG Feature Descriptor [24] for Handwritten Script Classification at Page, Block, Text-line, and Word Levels

Classifier	**Recognition Accuracy (%)**			
	Level of identification			
	Page level	**Block level**	**Text-line level**	**Word level**
Naïve Bayes	85.29	85.65	81.88	78.59
SVM	89.65	89.87	86.18	83.35
MLP	90.11	**91.88**	87.24	85.92
AdaBoost	86.73	86.54	82.39	78.48
Random Forest	87.16	87.59	84.87	82.29
Logistic Regression	85.33	86.17	83.65	80.17

(Best case is highlighted in bold).

Table 6. Individual Classifier Performance of GLCM Feature Set [25, 26] for Handwritten Script Classification at Page, Block, Text-line, and Word Levels

Classifier	**Recognition Accuracy (%)**			
	Level of identification			
	Page level	**Block level**	**Text-line level**	**Word level**
Naïve Bayes	85.57	83.39	81.27	83.89
SVM	88.25	86.47	88.18	85.05
MLP	89.33	88.89	**89.67**	85.58
AdaBoost	85.99	82.21	82.60	84.55
Random Forest	86.54	85.83	86.35	83.20
Logistic Regression	85.96	83.15	83.42	84.75

(Best case is highlighted in bold).

91.88% at the block level, 87.24% at the text-line level, and 85.92% at the word level. So, the highest recognition accuracy is found to be 91.88% at the block level. The detailed script-wise performance for the best case of the MLP classifier is revealed in Table S3 as separate supplementary material.

5.2.2 Using GLCM Feature Descriptor. The script identification performance of the GLCM feature descriptor [25, 26] is compared at page, block, text-line, and word levels for each individual classifier. The identification results are shown in Table 6. It is evident from Table 6 that the best-performing classifier is the MLP, which attains the identification accuracies of 89.33%, 88.89%, 89.67%, and 85.58% at page, block, text-line and word levels, respectively. As a result, the highest recognition accuracy is found to be 89.67% for the text-line level script identification. The detailed script-wise performance for the best case of MLP classifier is tabulated in Table S4 as separate supplementary material.

5.2.3 Using Combination of DWT and RT. The script identification performance of a combination of DWT- and RT-based features [27] are compared at page, block, text-line, and word levels for each individual classifier. The overall results are tabulated in Table 7. For page, block, text-line, and word level script identification, the MLP classifier achieves the highest identification accuracies of 88.50%, 88.28%, 88.87%, and 85.15%, respectively. It is also seen from Table 7 that the best recognition accuracy is found to be 88.87% at the text-line level. The detailed script-wise performance for the best case of the MLP classifier is shown in Table S5 as separate supplementary material.

Table 7. Individual Classifier Performance of the Combination of DWT- and RT-based Features [27] for Handwritten Script Classification at Page, Block, Text-line, and Word Levels

Classifier	Recognition Accuracy (%)			
	Level of identification			
	Page level	Block level	Text-line level	Word level
Naïve Bayes	88.21	85.19	86.24	81.66
SVM	90.25	88.83	89.97	85.58
MLP	89.50	88.28	**88.87**	85.15
AdaBoost	88.06	85.39	86.02	82.57
Random Forest	88.47	87.10	87.98	84.32
Logistic Regression	88.54	86.55	86.29	82.18

(Best case is highlighted in bold).

Table 8. Individual Classifier Performance of MLG Transform-based Features [28] for Handwritten Script Classification at Page, Block, Text-line, and Word Levels

Classifier	Recognition Accuracy (%)			
	Level of identification			
	Page level	Block level	Text-line level	Word level
Naïve Bayes	89.49	84.19	86.50	82.17
SVM	93.17	89.25	89.06	86.01
MLP	**95.83**	90.56	89.39	**87.75**
AdaBoost	89.75	87.94	86.42	84.26
Random Forest	91.55	89.08	88.86	86.15
Logistic Regression	91.21	88.41	88.35	86.67

(Best case is highlighted in bold).

5.2.4 Using MLG Transform. The script identification performance of the MLG transform based feature set [28] is compared at page, block, text-line, and word levels for each individual classifier. Table 8 illustrates the results achieved by each individual classifier at all the four levels. The MLP classifier performs the best among the six classifiers. Script recognition accuracies of 95.83%, 90.56%, 89.39%, and 87.75% are attained at page, block, text-line, and word levels, respectively. Consequently, the highest recognition accuracies are found to be 95.83% and 87.75% for the page level and word level script identification. The detailed script-wise performances for these best two cases of MLP classifier are displayed in Tables (S6–S7) and are provided as separate supplementary material.

5.2.5 Using Combination of NGTDM- and GLRLM-based Feature Set. The script identification performance of a combination of NGTDM- and GLRLM-based feature set [29] is compared at page, block, text-line, and word levels for each individual classifier and tabulated in Table 9. Table 9 shows that the highest identification accuracy is observed using the MLP classifier. The script recognition accuracies of 93.05%, 89.05%, 89.36%, and 84.94% are achieved by the MLP classifier at page, block, text-line, and word levels, respectively. So, the highest recognition accuracy is found to be 93.05% at the page level. The detailed script-wise performance for the best case of the MLP classifier is illustrated in Table S8 as separate supplementary material.

5.2.6 Using DHT Algorithm. The script identification performance of the DHT algorithm [30] is compared at page, block, text-line, and word levels for each individual classifier. The performance results are shown in Table 10. The best-performing classifier is the MLP classifier, which achieves

Table 9. Individual Classifier Performance of the Combination of NGTDM- and GLRLM-based Features [29] for Handwritten Script Classification at Page, Block, Text-line, and Word Levels

Classifier	Recognition Accuracy (%)			
	Level of identification			
	Page level	Block level	Text-line level	Word level
Naïve Bayes	85.50	83.34	85.95	79.10
SVM	92.83	89.56	89.70	84.62
MLP	**93.05**	89.05	89.36	84.94
AdaBoost	88.35	85.78	87.49	81.85
Random Forest	90.62	87.19	88.23	83.06
Logistic Regression	90.56	86.42	88.67	82.98

(Best case is highlighted in bold).

Table 10. Individual Classifier Performance of DHT Algorithm [30] for Handwritten Script Classification at Page, Block, Text-line, and Word Levels

Classifier	Recognition Accuracy (%)			
	Level of identification			
	Page level	Block level	Text-line level	Word level
Naïve Bayes	79.89	79.09	80.15	75.20
SVM	89.55	88.42	88.08	84.19
MLP	90.93	**90.11**	88.67	84.24
AdaBoost	83.60	81.28	85.39	80.35
Random Forest	88.21	86.55	87.58	84.06
Logistic Regression	86.04	84.86	88.45	83.75

(Best case is highlighted in bold).

the accuracies of 90.93%, 91.11%, 88.67%, and 84.24% for page, block, text-line, and word level script identification, respectively. Table 10 illustrates that the highest recognition accuracy is achieved by a MLP classifier, which is found to be 91.11% for the block level script identification. The detailed script-wise performance for the best case of the MLP classifier is presented in Table S9 as separate supplementary material.

5.3 Summarization of Benchmark Handwritten Script Classification Results

Table 11 summarizes the benchmark multi-script classification outcomes obtained on our in-house database at four different levels using MLP classifier for both structural-based and visual appearance-based features. Based on the recognition accuracies achieved by MLP classifier, different feature sets performed in a diverse manner at each level of identification and can be arranged accordingly. For page level, the script classification ability of different feature sets can be represented as: MLG Transform > NGTDM and GLRLM > DHT algorithm> HOG > DWT and RT > GLCM. At block level script classification, the performances of different feature sets can be arranged as: HOG > MLG Transform > DHT algorithm> NGTDM and GLRLM > GLCM > DWT and RT. Similarly, for text-line level script classification, the observation is: GLCM > MLG Transform > NGTDM and GLRLM > DWT and RT > DHT algorithm > HOG. Finally, for word level script classification, the order of performance is: MLG Transform > HOG > GLCM > DWT and RT > NGTDM and GLRLM > DHT algorithm. It is to be noted that in case of word level, the performance of visual appearance-based features is found to be much better than structural-based features.

Table 11. Summarization of the Performance Results Reported for Different Feature Sets Using MLP Classifier Measured at Different Identification Levels

Feature set	Recognition Accuracy (%) Level of classification				Average Recognition Accuracy (%)
	Page level	**Block level**	**Text-line level**	**Word level**	
Convex-hull-based features [19]	-	-	-	84.21	84.21
Elliptical and Polygonal Approximation-based features [20]	-	-	-	85.12	85.12
HOG [24]	90.11	**91.88**	87.24	85.92	88.79
GLCM [25, 26]	89.33	88.89	**89.67**	85.58	88.37
DWT and RT [27]	88.50	88.28	88.87	85.15	87.95
MLG Transform [28]	**95.83**	90.56	89.39	**87.75**	**90.88**
NGTDM and GLRLM [29]	93.05	89.05	89.36	84.94	89.10
DHT algorithm [30]	90.93	90.11	88.67	84.24	88.49

(Best recognition accuracy is styled in bold). The first two feature sets are structural-based, whereas the rest are visual appearance-based features.

It can also be observed from Table 11 that using MLG Transform, the highest recognition accuracies are found to be 95.83% and 87.75% for page level and word level script classifications, respectively. Similarly, using HOG feature descriptor, the best script classification accuracy of 91.88% is reported at block level. Again, at text-line level, the highest classification accuracy of 89.67% is achieved using GLCM feature set. Thus, highest classification accuracies of 95.83%, 91.88%, 89.67%, and 87.75% are received for page, block, text-line, and word level script identification, respectively. Tables (S10–S13) show the confusion matrices produced by the best cases of MLP classifier for page, block, text-line, and word level script classifications, respectively. Tables (S10–S13) are provided as supplementary materials with this manuscript. Based on the performance of the feature sets, the script classification performances at different levels can be ordered as: page level > block level > text-line level > word level. Thus, the final conclusion drawn on the basis of preceding experimentation is outlined below:

- Page level data is more stable and performs the best, irrespective of the features chosen.
- Performances of block and text-line level data are relatively similar.
- Handwritten script classification at word level is the worst among all the four levels.

It can be examined from the result analysis that visual appearance-based features performed much better than structural-based features. The reason is that structural-based features are not translation-, rotation-, and scale-invariant, at the same time very sensitive to boundary noise and variations. Furthermore, the number of primitives necessary for each shape is not known, as there is no proper definition for an object or shape. A variation of object boundaries causes variations to the primitives, so it is less consistent than visual appearance-based methods. Another major limitation of structural-based features is that they can only be applied for script classification at word level, whereas visual appearance-based features can be applied on any portion of given script text. Visual appearances-based methods effectively use all the pixel information within the script region, whereas structural-based features take into account only the shape contour, which may not be sometimes important for some applications. For example, structural-based features can classify between handwritten *Devanagari* (which is a Matra-based script) and *Roman* (which is a non-Matra-based script) scripts, since the shape variations of these two scripts are almost

Table 12. Recommendation of Different Feature Sets at Suitable Levels of Classification

Feature set	Feature dimension	Suitable level of classification	Highest Recognition accuracy (%)
Convex-hull-based features [19]	145	Word level	84.21
Elliptical and Polygonal Approximation features [20]	82	Word level	85.12
HOG [24]	80	Block level	91.88
GLCM [25, 26]	80	Text-line level	89.67
DWT and RT [27]	48	Text-line level	88.87
MLG Transform [28]	60	Page level	95.83
		Word level	87.75
NGTDM and GLRLM [29]	84	Page level	93.05
DHT algorithm [30]	72	Block level	90.11

diverse in nature. On the contrary, these types of features fail to classify between handwritten *Devanagari* and *Gurumukhi* scripts, since they are both Matra-based scripts, and the shape variations between these two scripts are almost similar in nature. In this case, visual appearance-based methods can be very effective, because they take the whole script region into consideration for shape representation and description. Based on the overall classification accuracies averaged over all identification levels, the performances of visual appearance-based features can also be arranged sequentially. The order of their performance is: MLG Transform > NGTDM and GLRLM > HOG > DHT algorithm > GLCM > DWT and RT. So, it can be noticed that MLG Transform is the most consistent and discriminatory feature set, as it shows the best averaged performance at any level of classification. However, the combination of DWT and RT is the worst performer among all the feature sets. With respect to the suitability of feature set(s) applied at a particular level of identification, it can be recommended that the MLG Transform is the most suited for both page and word level script classification, whereas HOG and GLCM feature descriptors are appropriate for block level and text-line level script classification, respectively. Table 12 depicts the suitable level of identification for a given feature set.

In this case study, a hierarchical classification methodology based on a tree structure (having three levels) has also been implemented as an alternative solution. The three-level tree architecture for handwritten script identification of 12 official *Indic* scripts is illustrated in Figure 10. The hierarchy is designed in such a way that it allows us to cluster the scripts with some common visual characteristics as a node at one level and then focus on the classification based on some other visual feature of the scripts at the next level. This, in turn, will lead to model an improved classification method. The grouping of the scripts at each level of the tree structure is based on some *inter-script* specific features. Finally, each group of scripts is classified based on unique *intra-script* distinctive features. Based on the data of the *confusion matrix* (illustrated in Tables S10–S13 as supplementary materials) with a bias to the *visual appearances* of the scripts [54], it is inferred that the Matra-based and non-Matra-based scripts could be formed two clusters of different scripts. So, at Level-1, all the 12 scripts are assembled into two sub-groups, i.e., "**G1:** *Gurumukhi, Devanagari, Manipuri, Bangla*, and **G2:** *Gujarati, Oriya, Telugu, Kannada, Tamil, Malayalam, Urdu* and *Roman.*" The feature extraction for this classification is done using DHT algorithm At Level-2, the feature descriptors based on MLG Transform are applied for the *intra-script* classification belonging to **G1**. However, the group **G2** is again classified into four distinct sub-groups, i.e., **G2.1:** *Gujarati, Oriya*; **G2.2:** *Kannada, Malayalam, Tamil, Telugu*, **G2.3:** *Urdu,* and **G2.4:** *Roman.* This classification is also done using MLG-based feature descriptor. At the final level, i.e., Level-3, the two scripts (*viz.,*

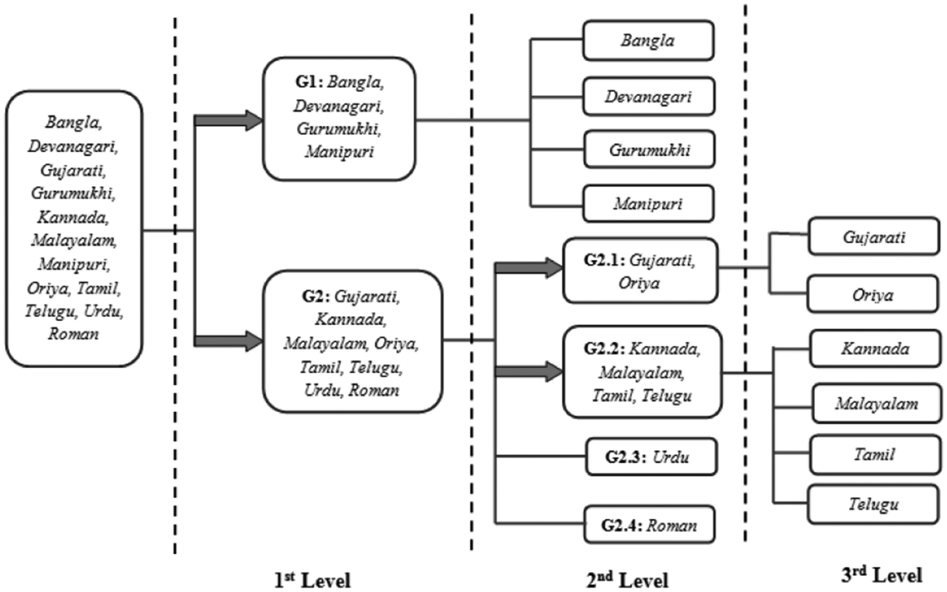


Fig. 10. Diagrammatic representation of the hierarchical three-level tree-based architecture for handwritten script classification problem.

Fig. 11. Graph showing the classification accuracies attained by using hierarchical tree-based architecture for handwritten script classification problem at four different levels.

Gujarati and *Oriya*) in sub-group **G2.1** are finally classified. On the contrary, the four South-*Indic* scripts under the second sub-group, **G2.2,** are also identified based on MLG Transform. The average classification accuracies achieved by using this tree-based methodology at page level, block level, text-line-level, and word level are shown with the help of a bar chart, illustrated in Figure 11.

5.4 Statistical Significance Test

It is quite evident from the previous subsection that the MLG Transform is the most suited for both page and word level script classification, whereas HOG and GLCM feature descriptors are appropriate for block level and text-line level script classification, respectively. In this subsection,

Table 13. Summarization of Results of Statistical Significance Test for Handwritten Script Classification Computed at Page, Block, Text-line, and Word Levels

Level of classification	**Degrees of Freedom**	**Level of significance**	**Critical value**	**Calculated value**	**Null Hypothesis (Accepted/Rejected)**
Page level	5	0.05	11.07	27.057	Rejected
Block level				25.47	
Text-line level				26.63	
Word level				39.73	

the statistical significance test of the present experimental setup is carried out to validate the performance of the multiple classifiers using multiple datasets. The statistical significance test is performed to validate the best-performing feature descriptors at four different levels of script classification. A comprehensive description of this test is already presented in [55]. Here, the null hypothesis states that there is no significant difference among the classification abilities of the six classifiers considered here. Before performing this test, page level, block level, text-line level, and word level datasets are *randomly* divided to create small subsets with different *sample sizes*. Performances of six classifiers are carried out for each of these *randomly* created subsets. A safe and robust non-parametric Friedman test [56, 57] is then performed for validating the performances of the multiple classifiers using multiple datasets at four different levels of handwritten script classification. Tables (S14–S17) depict the performances (in terms of classification accuracies) as well as their assigned ranks (required for performing Friedman test) achieved by six classifiers on 12 *randomly* chosen page, block, text-line, and word levels datasets, respectively, which are provided as supplementary materials. The overall results of the Friedman test for all the four levels of script classification are summarized in Table 13. It can be noted from Table 13 that the Friedman statistic rejects the null hypothesis. Hence, it can be said that there exist significant differences among the classification abilities of the six classifiers, which, in turn, statistically validates our performance results.

5.5 Variation of Block Sizes for Block Level Script Classification

Since HOG descriptor [24] scores the highest recognition accuracy on block level script datasets, we have carried out more experiments on block level datasets by varying the block-size of the handwritten text images in different scripts. The overall results are recorded as a graph shown in Figure 12. Here, we have chosen five different block sizes such as 64×64, 128×128, 256×256, 512×512, and $1,024 \times 1,024$. It can be understood from Figure 11 that the SVM classifier attains the best recognition accuracies of 87.17% and 88.06% when the block sizes of text images are 64×64 and 128×128, respectively. Moreover, MLP classifier scores the highest accuracies of 91.88%, 92.98%, and 93.2% for block sizes of 256×256, 512×512, and $1,024 \times 1,024$, respectively. It is also clear from the results that the script recognition accuracies gradually increase with the size of text blocks.

5.6 Variation of Training and Testing Samples of Script Datasets

The next experiment is done by varying the train and test sizes of page level, block level, text-line level, and word level datasets and observing the effect of this variation when six different visual appearance-based feature sets are applied. The visual appearance-based feature sets perform better than structural-based features. This is the reason for considering only the former feature sets for our experimentation. We have considered five different cases as described here. Case I: training set consists of 90% of handwritten script datasets at four different levels and test set consists of 10%

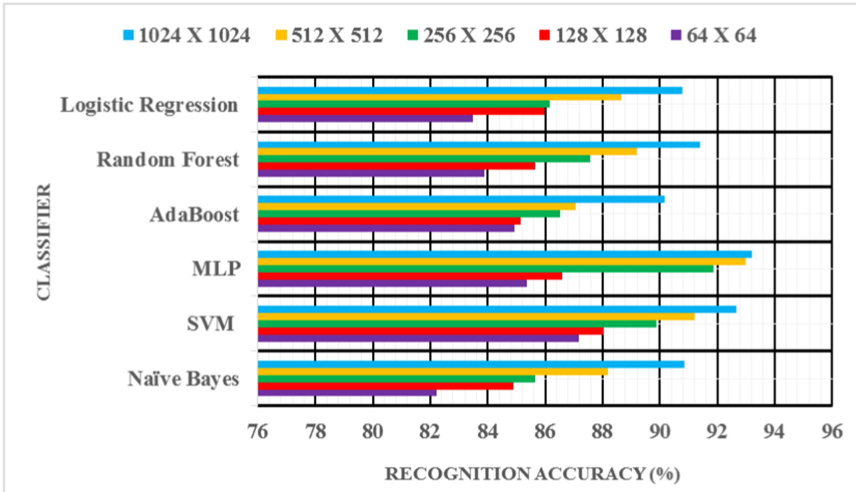


Fig. 12. Graph showing the recognition accuracies scored by six different classifiers on varying block sizes for block level script datasets.

of handwritten script datasets at four different levels; Case II: training size, 70%, and test size, 30%; Case III: training size, 50%, and test size, 50%; Case IV: training size, 30%, and test size, 70%; Case V: training size, 10%, and test size, 90%. The classification of the script datasets is done using MLP classifier, which performs the best in almost all occasions. Table 14 shows the performance results of the classifier for these five different cases. For the first case, MLG transform performs the best at all the four levels. In second case, MLG transform scores the highest classification accuracies at both page and word levels, whereas HOG and GLCM feature descriptors show the highest classification accuracies in case of block level and text-line level datasets, respectively. Furthermore, in the third case, MLG transform, DHT algorithm, DWT and RT, NGTDM and GLRLM feature descriptors show the best accuracies on page level, block level, text-line level, and word level datasets, respectively. Similarly, in case IV, MLG transform achieves the highest classification accuracies at both page and word levels. At block and text-line levels, DHT algorithm and GLCM feature descriptor perform the best among all feature descriptors. Finally, in case V, MLG transform, HOG, DWT and RT, NGTDM, and GLRLM feature descriptors attain the best accuracies at page level, block level, text-line level, and word level, respectively.

5.7 Effect of Feature Dimensionality Reduction on Script Classification Performances

As it is evident from Section 5.3 that visual appearance-based features perform better than the structural-based features. Keeping this mind, in this subsection, an additional experimentation is done by applying two previously proposed feature selection methods after combining all the feature sets extracted using six different visual appearance-based features, namely, HOG, GLCM, DWT and RT, MLG transform, NGTDM and GLRLM, and DHT algorithm. It is to be noted from Table 3 that the size of the original feature vector is found to be 484 after merging all the feature sets mentioned above. Feature selection is a useful procedure for selecting the optimal feature subset (or reducing the size of feature dimension) by removing the redundant features in the original feature space. This, in turn, helps to decrease the overall training time and increase the classification accuracy. In the present work, we have used two feature dimensionality reduction techniques named **Principal Component Analysis (PCA)** [58] and **Harmony search (HS)**-based feature

Table 14. Recognition Accuracies Scored by MLP Classifier Using Six Different Visual Appearance-based Feature Sets for Five Different Cases Varying the Training and Test Sets

Feature Set	Level of classification	Recognition accuracy (%)				
		Case I	Case II	Case III	Case IV	Case V
HOG [24]	Page level	93.13	90.11	89.53	87.74	83.5
	Block level	92.87	**91.88**	88.16	86.15	**83.75**
	Text-line level	90.30	87.24	86.67	84.33	79.12
	Word level	88.42	85.92	83.20	80.56	75.97
GLCM [25, 26]	Page level	92.75	89.33	88.55	86.60	81.14
	Block level	90.83	88.89	86.91	84.24	79.48
	Text-line level	91.36	**89.67**	88.73	**85.85**	80.07
	Word level	88.97	85.58	83.38	80.02	74.55
DWT and RT [27]	Page level	91.62	88.50	87.55	85.48	80.10
	Block level	90.85	88.28	87.33	85.16	80.98
	Text-line level	90.17	88.87	**87.90**	84.57	**80.33**
	Word level	87.24	85.15	82.92	78.65	75.42
MLG transform [28]	Page level	**98.33**	**95.83**	**95.04**	**94.30**	**90.02**
	Block level	**93.70**	90.56	88.65	86.52	81.64
	Text-line level	**92.91**	89.39	86.17	85.74	79.27
	Word level	**91.65**	**87.75**	82.04	**81.58**	72.26
NGTDM and GLRLM [29]	Page level	95.53	93.05	92.25	91.8	85.06
	Block level	91.92	89.05	88.33	86.34	80.78
	Text-line level	90.45	89.36	87.18	84.69	78.83
	Word level	87.24	84.94	**85.75**	78.85	**75.94**
DHT algorithm [30]	Page level	92.75	91.93	89.50	87.16	82.52
	Block level	92.33	90.11	**89.58**	**88.42**	80.18
	Text-line level	90.16	88.67	86.42	84.98	80.05
	Word level	86.09	84.24	83.86	81.63	75.30

(best accuracies for each case and at each level are marked in bold).

selection method [59] for page level, block level, text-line level, and word level handwritten script classification. The overall results obtained (using the best-performing MLP classifier) by both feature dimensionality reduction methods are given in Table 15. It is evident from Table 15 that HS-based feature selection method performs better than PCA by selecting lesser number of features and giving higher classification accuracy. For all the four levels of script classification, PCA produces an improvement of about 2%–3% in the overall classification accuracy while selecting 82% of the original number of features. However, the HS-based method produces about 5%–6% increment in the overall classification accuracy with only 70% of the original number of features. This proves the effectiveness and efficiency of using feature selection/feature dimensionality reduction methods for handwritten script classification problem.

5.8 Performance Evaluation on Freely Available Dataset

The literature survey carried out in Section 2 describes that there is only one benchmark database available for handwritten script classification problem, which is named as *PHD_Indic11* dataset developed by Obaidullah et al. [60]. They used a combination of two types of feature sets, namely, script-dependent (containing 56 feature values) and script-independent (having 56 feature values) feature sets and achieved script recognition accuracies of 98%, 98.75%, 97.50%, and 93.93% at page

Table 15. Script Classification Performance (in Terms of Number of Selected Feature and Recognition Accuracies) Obtained by Applying Feature Dimensionality Reduction Methods and Using MLP Classifier for the Combination of Visual Appearance-based Feature Sets

Level of classification	Size of original feature descriptor	Original recognition accuracy (%)	PCA [58]		HS-based method [59]	
			Number of optimal features selected	Recognition accuracy (%)	Number of optimal features selected	Recognition accuracy (%)
Page level	484	92.78	385	95.13	314	97.44
Block level		90.65	392	91.87	326	95.67
Text-line level		88.875	405	91.25	342	94.89
Word level		84.02	411	86.97	369	90.58

Table 16. Recognition Performances Reported for Different Feature Sets Using MLP Classifier Measured at Different Identification Levels on *PHD_Indic11* Dataset Proposed in Reference [60]

Feature set	Recognition accuracy (%)				Average recognition accuracy (%)
	Level of classification				
	Page level	**Block level**	**Text-line level**	**Word level**	
Convex-hull-based features [19]	-	-	-	86.52	86.52
Elliptical and Polygonal approximation-based features [20]	-	-	-	87.64	87.64
HOG [24]	93.5	95.18	98.56	90.60	94.96
GLCM [25, 26]	92.00	89.92	**98.20**	92.98	93.27
DWT and RT [27]	90.34	91.55	94.89	92.55	92.33
MLG Transform [28]	**98.60**	94.84	97.28	**94.04**	**96.19**
NGTDM and GLRLM [29]	95.45	93.26	95.06	93.12	94.22
DHT algorithm [30]	93.12	**97.89**	97.45	93.86	95.58

(Best Recognition Accuracy is Styled in Bold).

level, block level, text-line level, and word level, respectively. In our case study, the structural-based and visual appearance-based feature descriptors are made to run on this script datasets. The overall results are tabulated in Table 16. It can be seen from Table 16 that the MLG Transform scores the highest classification accuracies of 98.6% and 94.04% at page level and word level datasets, respectively. DHT algorithm and GLCM feature descriptor score the highest accuracies of 97.89% and 98.2% in case of block level and text-line level datasets, respectively. However, it can be noticed that MLG transform is the most consistent feature set, as it shows the best averaged performance at any level of script classification. However, the combination of DWT and RT does not perform well.

Besides these, the authors Ukil et al. achieved an overall word level accuracy of 94.73% on *PHD_Indic11* dataset with a deep learning approach [76] using a set of 10 different **Convolutional Neural Networks (CNNs)** comprising a set of 10,240 features. Again, in the work done by Ukil et al. in Reference [77], the authors attained 95.04% on the same dataset word level accuracy by combining 12 small integrated CNNs models each having a feature dimension of 1,024 elements. This implies that the average accuracy achieved by using machine learning approaches is found to be about 1% less than that of deep learning approaches. This is negligible, considering the resource and time complexities associated with the deep learning approaches.

6 FUTURE SCOPE OF HANDWRITTEN *INDIC* SCRIPT CLASSIFICATION RESEARCH

The problem of *Indic* script classification was identified nearly two decades ago. Few techniques are also developed to solve this problem. There exist a lot of works that can arise as future scope from handwritten script classification. Some of them are listed below as follows:

6.1 Availability of Benchmark Multi-script Databases

It is evident from the literature survey that only one *freely available* multi-script database (known as *PHD_Indic11* dataset [60]) exists that can be used for evaluating any handwritten *Indic* script classification algorithm. It can be observed that in different domains such as handwritten digit recognition, scene text detection, face recognition, and so on, numerous *freely available* benchmark datasets with varying sizes exist in the literature. So, to meet the need of this domain, researchers should come up with more benchmark databases for handwritten *Indic* script classification, and these databases should be made *freely available* to the research community. A survey presented by Hussain et al. [61] reports the benchmark databases developed in the field of handwritten document analysis.

However, in future releases of our in-house database, the database quantity can be increased, which involves collection of more multi-script handwritten document pages containing purely *Indic* scripts. This will include more variations of writing styles that, in turn, will provide a more realistic assessment of the handwritten script classification algorithms in multi-script scenario. The database compilation may also include mixed-script document pages written in *Indic* scripts mixed with *Roman* or other scripts, which itself is also a less-explored domain.

6.2 Need of Feature Selection Methods

Researchers may be encouraged to come up with some feature selection methods in an effort to identify more informative and discriminative feature subsets. Application of nature-inspired stochastic search techniques generates multiple good-quality feature subsets without resorting to exhaustive search. The concept of feature selection is already implemented for typical pattern recognition problems such as speech recognition [62], face recognition [63, 64], handwritten digit recognition [65], text classification [66], handwritten word recognition [67], gene selection in microarray data [68], and so on. Handwritten script classification techniques can benefit greatly if intelligent feature selection can be used to remove the noisy, irrelevant, redundant, or misleading features, which will enhance the accuracy of the recognition system. The work described by Singh et al. [59] has already investigated the HS-based optimization algorithm for selection of optimal feature subset in handwritten script classification problem. This study justifies the need of feature selection for handwritten script classification, where both local and global features are included, without knowing the exact importance of features. Although the results show a quite encouraging trend, much work (as reported in Reference [69]) can still be done to further increase the performance results.

6.3 Deep Learning-based Approaches

Nowadays, deep learning-based approach has become a successful alternative to traditional machine learning-based approach. Deep learning models such as deep belief network, deep neural network, convolutional neural network, and recurrent neural network have been applied to diverse research fields, including speech recognition, computer vision, medical image analysis, audio recognition, machine translation, social network filtering, bioinformatics, and natural

Fig. 13. Sample images of scene text detection in multi-script environment.

language processing, where the outcomes are found to be comparable to and in some cases superior to human experts [70–75]. In recent times, a few works described by Ukil et al. in References [76, 77] and Kundu et al. in Reference [78] already introduced deep learning approaches for solving the problem of handwritten script classification. Application of deep learning-based approaches should be explored much in the near future.

6.4 Demand for Video-based Script Classification

Script classification from videos is almost an uncharted research area as compared to that from printed and handwritten documents. This problem is also one of the challenging tasks that finds its application in scene understanding, machine translation, content-based video indexing, understanding of visual contents and retrieval, and so on. Reviewing the literature study, it is found that researchers Shivakumara et al. [79], Malik et al. [80], Bhunia et al. [81], Sharma et al. [82], and Gaikwad et al. [83] have developed methods that focus on classification of the scripts from videos. Considering the few works done in this domain, video-based script classification still has a long way to go in the research field.

6.5 Multi-script Scene Text Detection and Recognition

The problem of scene text detection can be defined as the procedure of detecting the text components in natural images. This problem is challenging due to variations or diversities (different fonts, sizes, and orientations) of texts in the wild, complex backgrounds, presence of noise or blur due to low light conditions and so on. Scene text recognition has wide applications in our day-to-day life activities, such as identification of vehicles by reading the number plate, recognizing sign-boards, indexing of multimedia, and so on. A recent survey by Lin et al. [84] indicates that a lot of work [85–100] has been performed by researchers for solving this task of scene text detection and recognition. But in multilingual countries such as India, USA, Malaysia, South Africa, Singapore, and so on, scene text images often contain text printed in two or more scripts. This problem is known as multi-script scene text detection and recognition. Figure 13 shows this situation. In these cases, the scripts must be known beforehand for the recognition of text in such images, which gives rise to the problem of scene text recognition in multi-script environment. However, scene text recognition in a multi-script environment has been reported in some recent works described in References [101–105]. But one of the major limitations of these works is that researchers considered only a few scripts. Therefore, it can be said that a comprehensive model for multi-script scene text detection and recognition is still in its infancy stage.

6.6 Script-independent Writer Identification/Verification

The problem of writer identification can be defined as automatic identification of the writer of a handwritten text document from a given set of handwriting samples based on the fact that the writing sample of the same writer is present in the database [106]. Writer verification is the problem of authenticating a handwritten text document whether the same is written by that individual writer or not. Writer identification/verification is an important research topic due to its large number of applications in forgery recognition, forensic sciences, criminology, organizations dealing with historical documents, and so on. The works done to date reveal that researchers have considered the problem of writer identification/verification for text documents written in *Latin* [107], *Gurumukhi* [108], *Kannada* [109, 110], *Bangla* [111], *Telugu* [110, 112], and *Odia* [113] scripts only. However, the solution to this problem in a multi-script environment is still left unsolved and thus becomes an important future scope to handwritten script classification.

6.7 ICDAR Robust Reading Competition

Robust reading refers to the research area that helps to bridge the gap between the handwritten document analysis community and the wider computer vision community. This robust reading competition was first started in the year 2003 [114] by International Conference of Document Analysis and Recognition (ICDAR) and continued in 2005 [115], 2011 [116, 117], 2013 [118], 2015 [119], and 2017 [120–122]. The competition is arranged in the form of challenges signifying explicit application domains for robust reading. These challenges are chosen in such a way that they can include a wide-ranging real-life state of affairs. In the coming years, these competitions involving cases of multi-script scenarios will serve as a footstep for conducting and comparing further research activities on a common platform.

6.8 Online *Indic* Script Classification

Online character recognition has gained a lot of popularity in the recent years. The growth of devices such as computers and smart phones can effectively interpret and digitize the data entered, which, in turn, increases the demand for online handwritten recognition. Research works for online character recognition have already been performed for some major *Indic* scripts like *Bangla* [123, 124], *Telugu* [125], *Devanagari* [126–129], *Tamil* [126, 130, 131], *Gurumukhi* [132–134], *Kannada* [135–137], *Malayalam* [138–140], *Urdu* [141–143], *Gujarati* [144–146], and so on. However, the problem of online script classification considering all the *Indic* scripts together is missing in the literature. It is to be noted that all the works described in this case study are related to offline recognition of *Indic* scripts, but the recognition of scripts in online mode is still yet to be exposed by the researchers.

7 CONCLUSION

A pre-OCR script classification is an important aspect for any multi-script/multilingual country. Due to a variety of applications, handwritten script recognition/classification is gaining more importance in today's electronically interconnected society. In the present work, a case study for designing a comprehensive offline handwritten *Indic* script classification system, which considers input text sample as a whole document page or a text block or a text-line or a simple word from the document page, has been undertaken. The main objective in this case study is to address eight major concerns: (1) preparing handwritten script database written in 12 constitutionally enlisted Indian scripts at the above-mentioned four different levels; (2) measuring the individual performances of handwritten script classification system considering the input scripts from page, block, text-line, and word level images; (3) selecting an ideal level for handwritten script classification among the above-mentioned four levels; (4) performance measurements of different feature sets at four different script levels that have been evaluated and statistically validated using six state-of-the-art classifiers such as Naïve Bayes, SVM, MLP, AdaBoost, Random Forest, and Logistic Regression; (5) detailed performance measures of the best case of handwritten script classification results; (6) improving the overall script classification performance by using feature dimensionality reduction methods; (7) on two different handwritten *Indic* script databases, of which one is an in-house developed dataset and the other one is a freely available dataset; and (8) scope of future research works that may be carried out as an application to handwritten *Indic* script classification.

For experimentation, a handwritten script database consisting of page, block, text-line, and word level datasets is prepared. A set of eight different features consisting of two structure-based features and six visual appearance-based features are considered for the feature extraction purpose. The structure-based features comprise Convex-hull-based features and a combination of Elliptical and Polygonal Approximation-based features, whereas the visual appearance-based features include HOG, GLCM, combination of DWT and RT, MLG transform, NGDTM and GLRLM, and DHT algorithm. The former feature set is applied for word level script identification only, whereas the latter feature set is applied and compared at each level of script identification. Six well-known machine learning algorithms are used for script classification purpose and among them the MLP classifier gives the best performance in nearly all circumstances. The proposed system is also tested on a freely available handwritten script classification database named *PHD_Indic11,* and we have achieved better performance results than the state-of-the-art methods proposed in Reference [60].

To find the solution with optimum performance, it is noticed that the page level dataset achieves the highest recognition accuracy as compared to block, text-line, and word levels. It is meant that the classification accuracy of script recognition decreases simultaneously with the decrease in size of the available information extracted from the document page. With respect to the feature set, it is found that visual appearance-based features are well suited at all levels of script classification than structure-based features. Regarding visual appearance-based features, it is noted that MLG transform accomplishes much better classification results among all, followed by NGTDM and GLRLM, HOG, DHT algorithm, GLCM, DWT, and RT. While comparing the optimal feature set at each level, it is examined that MLG transform is best suited for both page and word level script classification, whereas HOG and GLCM feature descriptors show the best performance for block level and text-line level script classifications, respectively. The highest average script classification accuracies at page, block, text-line, and word levels are found to be 95.83%, 91.88%, 89.67%, and 87.75%, respectively. As discussed in the previous section, it is quite clear from this case study that the comprehensive solution to this very problem still has a long way to go in the near future. For example, some classifier combination approaches reported in References [147, 148] are also needed for handwritten *Indic* script classification to improve the overall classification accuracy.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this article.

REFERENCES

[1] D. Ghosh, T. Dube, and A. P. Shivaprasad. 2010. Script Recognition: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 12 (2010), 21422161.

[2] Sk. Md. Obaidullah, K. C. Santosh, N. Das, C. Halder, and K. Roy. 2018. Handwritten Indic Script Identification in Multi-script Document Images—A Survey. *Int. J. Pattern Recogn. Artif. Intel.* 32, 10 (2018), 1856012.

[3] S. Kaur, S. Bawa, and R. Kumar. 2019. A survey of mono- and multi-lingual character recognition using deep and shallow architectures: Indic and non-Indic scripts. *Artif. Intell. Rev.* DOI: https://doi.org/10.1007/s10462-019-09720-9

[4] Sk. Md. Obaidullah, C. Halder, K. C. Santosh, N. Das, and K. Roy. 2018. Automatic line-level script identification from handwritten document images—A region-wise classification framework for Indian subcontinent. *Malays. J. Comput. Sci.* 31, 1 (2018), 63–84.

[5] M. Ferrer, A. Morales, U. Pal, et al. 2013. LBP based line-wise script identification. In *12th IEEE International Conference on Document Analysis and Recognition (ICDAR).* 369–373.

[6] S. Jetley, K. Mehrotra, A. Vaze, and S. Belhe. 2014. Multi-script identification from printed words. In *Image Analysis and Recognition.* A. Campilho and M. Kamel (Eds.) *(LNCS,* vol. 8814). Springer, 359–368.

[7] L. Mioulet, U. Garain, C. Chatelain, P. Barlas, and T. Paquet. 2015. Language identification from handwritten documents. In *13th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. 676–680.
[8] L. Shijian and C. L. Tan. 2008. Script and language identification in noisy and degraded document images. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 1 (2008), 14–24.
[9] M. Kumar, M. Jindal, R. Sharma, and S. R. Jindal. 2018. Character and numeral recognition for non-Indic and Indic scripts: A survey. *Artif. Intell. Rev.* 1–27. Q5
[10] M. K. Sharma and V. P. Dhaka. 2016. Pixel plot and trace based segmentation method for bilingual handwritten scripts using feedforward neural network. *Neural Comput. Applic.* 27, 7 (2016), 1817–1829.
[11] A. Ul-Hasan, M. Afzal Zeshan, F. Shafait, M. Liwicki, and T. Breuel M. 2015. A sequence learning approach for multiple script identification. In *13th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. 1046–1050.
[12] S. Kavitha, P. Shivakumara, G. H. Kumar, and C. Tan. 2015. A robust script identification system for historical Indian document images. *Malay. J. Comput. Sci.* 28, 4 (2015), 283–300.
[13] http://rajbhasha.nic.in/hi/languages_included_in_the_eighth_schedule. Q6
[14] G. G. Rajput and S. B. Ummapure. 2017. Line-wise script identification from handwritten document images using sift method. In *International Conference on Intelligent Computing Systems (ICICS)*. 117–125. Retrieved from https://ssrn.com/abstract=3126650.
[15] P. K. Singh, R. Sarkar, and M. Nasipuri. 2015. Offline script identification from multilingual Indic-script documents: A state-of-the-art. *Comput. Sci. Rev.* 15, C (2015), 1–28.
[16] P. Sahare and S. B. Dhok. 2017. Script identification algorithms: A survey. *Int. J. Multim. Inf. Retr.* 6, 3 (2017), 211–232.
[17] K. Ubul, G. Tursun, A. Aysa, D. Impedovo, G. Pirlo, and T. Yibulayin. 2017. Script identification of multi-script documents: A survey. In *IEEE Access* 5 (2017) 6546–6559. DOI: 10.1109/ACCESS.2017.2689159
[18] A. Kacem and A. Saïdani. 2017. A texture-based approach for word script and nature identification. In *Pattern Anal. Applic.* 20, 4 (2017), 1157–1167.
[19] P. K. Singh, R. Sarkar, N. Das, S. Basu, and M. Nasipuri. 2013. Identification of Devnagari and Roman script from multiscript handwritten documents. In *5th International Conference on Pattern Recognition and Machine Intelligence (PReMI) (LNCS, vol. 8251)*. 509–514.
[20] P. K. Singh, R. Sarkar, M. Nasipuri, and D. Doermann. 2015. Word level script identification for handwritten Indic scripts. In *13th IEEE International Conference on Document Analysis and Recognition.* 1106–1110.
[21] Sk. Md. Obaidullah, C. Goswami, K. C. Santosh, C. Halder, N. Das, and K. Roy. 2017. Separating Indic scripts with "Matra" for effective handwritten script identification in multi-script documents. *Int. J. Pattern Recogn. Artif. Intell.* 31, 5 (2017), 17.
[22] Sk. Md. Obaidullah, K. C. Santosh, C. Halder, N. Das, and K. Roy. 2019. Automatic Indic script identification from handwritten documents: Page, block, line and word level approach. *Int. J. Mach. Learn. Cybern.* 10, 1 (2019), 87–106.
[23] Sk. Md. Obaidullah, A. Bose, H. Mukherjee, K. C. Santosh, N. Das, and K. Roy. 2018. Extreme learning machine for handwritten Indic script identification in multi-script documents. *J. Electron. Imag.* 27, 5 (2018), 051214.
[24] P. K. Singh, A. Mondal, S. Bhowmik, R. Sarkar, and M. Nasipuri. 2014. Word level script identification from multi-script handwritten documents. In *3rd International Conference on Frontiers in Intelligent Computing Theory and Applications (FICTA)*.
[25] P. K. Singh, R. Sarkar, and M. Nasipuri. 2015. Line-level script identification for six handwritten scripts using texture based features. In *2nd Information Systems Design and Intelligent Applications.*
[26] P. K. Singh, S. K. Dalal, R. Sarkar, and M. Nasipuri. 2015. Page level script identification from multi-script handwritten documents. In *3rd IEEE International Conference on Computer, Communication, Control and Information Technology (C3IT)*. DOI: 10.1109/C3IT.2015.7060113
[27] P. K. Singh, S. Das, R. Sarkar, and M. Nasipuri. 2017. A two-stage approach to handwritten Indic script identification. In *Hybrid Intelligence for Image Analysis and Understanding*, S. Bhattacharyya, I. Pan, A. Mukherjee, and P. Dutta (Eds.). Wiley Publishers, 47–77.
[28] P. K. Singh, I. Chatterjee, and R. Sarkar. 2015. Page level handwritten script identification using modified log-Gabor filter based features. In *2nd IEEE International Conference on Recent Trends in Information Systems (ReTIS)*. 225–230.
[29] P. K. Singh, R. Sarkar, and M. Nasipuri. 2016. Statistical textural features for text-line level handwritten Indic script identification. In *3rd International Doctoral Symposium on Applied Computation and Security Systems (ACSS)*. 139–151.
[30] P. K. Singh, S. Das, R. Sarkar, and M. Nasipuri. 2016. Line parameter based word level Indic script identification system. *Int. J. Comput. Vis. Image Process.* 6, 2 (2016), 18–41.
[31] M. Hangarge, K. C. Santosh, and R. Pardeshi. 2013. Directional discrete cosine transform for handwritten script identification. In *12th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. 344–348.

[32] R. Pardeshi, B. B. Chaudhuri, M. Hangarge, and K. C. Santosh. 2014. Automatic handwritten Indian scripts identification. In *14th IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 375–380.

[33] Sk. Md. Obaidullah, C. Halder, N. Das, and K. Roy. 2015. Indic script identification from handwritten document images—An unconstrained block level approach. In *2nd IEEE International Conference on Recent Trends in Information Systems (ReTIS)*. 213–218.

[34] Sk. Md. Obaidullah, K. C. Santosh, Chayan Halder, Nibaran Das, and Kaushik Roy. 2017. Word level multi-script Indic document image dataset and baseline results on script identification. *Int. J. Comput. Vis. Image Process.* 7, 2 (2017), 81–94

[35] M. B. Vijayalaxmi and B. V. Dhandra. 2015. Script recognition using GLCM and DWT features. *Int. J. Adv. Res. Comput. Commun. Eng.* 4, 1 (2015), 256–260.

[36] Y. Mingqiang, K. Kidiyo, and R. Joseph. 2008. A survey of shape feature extraction techniques. In *Pattern Recognition, Peng-Yeng Yin* (Ed.). 43–90.

[37] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*. 886–893.

[38] R. M. Haralick, K. Shanmungam, and I. Dinstein. 1973. Textural features of image classification. In *IEEE Trans. Syst., Man, Cybern.* 3, 6 (1973), 610–621.

[39] R. C. Gonzalez and R. E. Woods. 1992. *Digital Image Processing I*. Prentice-Hall, India.

[40] N. Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Syst., Man., Cybern.* 9, 1 (1979), 62–66.

[41] J. G. Daugman. 1985. Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Optical Soc. Amer.* 2, 7 (1985), 1160–1169.

[42] M. Amadasun and R. King. 1989. Textural features corresponding to textural properties. *IEEE Trans. Syst., Man, Cybern.* 19, 5 (1989), 1264–1274.

[43] M. M. Galloway. 1975. Texture analysis using gray level run lengths. *Comput. Graph. Image Process.* 4 (1975), 172–179.

[44] A. Chu, C. M. Sehgal, and J. F. Greenleaf. 1990. Use of gray value distribution of run lengths for texture analysis. *Pattern Recogn. Lett.* 11, 6 (1990), 415–420.

[45] B. R. Dasarathy and E. B. Holder. 1991. Image characterizations based on joint gray-level run-length distributions. *Pattern Recogn. Lett.* 12, 8 (1991), 497–502.

[46] A. Khandelwal, P. Choudhury, R. Sarkar, S. Basu, M. Nasipuri, and N. Das. 2009. Text-line segmentation for unconstrained handwritten document images using neighborhood connected component analysis. In *3rd International Conference on Pattern Recognition and Machine Intelligence (PReMI) (LNCS, vol. 5909)*. 369–374.

[47] P. K. Singh, S. P. Chowdhury, S. Sinha, S. Eum, and R. Sarkar. 2016. Page-to-word extraction from unconstrained handwritten document images. In *1st International Conference on Intelligent Computing and Communication (ICIC2)*. 517–524.

[48] I. Rish. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*.

[49] C. Cortes and V. N. Vapnik. 1995. Support-vector networks. *Mach. Learn.* 20, 3 (1995), 273–297.

[50] K. Roy, C. Chaudhuri, M. Kundu, M. Nasipuri, and D. K. Basu. 2005. Comparison of the multilayer perceptron and the nearest neighbor classifier for handwritten digit recognition. *J. Inf. Sci. Eng.* 21, 6 (2005), 1245–1257.

[51] R. E. Schapire and Y. Singer. 1999. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.* 37, 3 (1999), 297–336, DOI: https://doi.org/10.1023/A:1007614523901

[52] T. K. Ho. 1995. Random decision forests. In *3rd IEEE International Conference on Document Analysis and Recognition* 278–282.

[53] S. Dreiseitl and L. Ohno-Machado. 2002. Logistic regression and artificial neural network classification models: A methodology review. *J. Biomed. Inform.* 35 (2002), 352–359.

[54] P. K. Singh, R. Sarkar, V. Bhateja, and M. Nasipuri. 2018. A comprehensive handwritten Indic script recognition system: A tree-based approach. *J. Amb. Intell. Human. Comput.* DOI: https://doi.org/10.1007_s12652-018-1052-4

[55] P. K. Singh, R. Sarkar, and M. Nasipuri. 2015. Statistical validation of multiple classifiers over multiple datasets in the field of pattern recognition. *Int. J. Appl. Pattern Recogn.* 2, 1 (2015), 1–23.

[56] M. Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Amer. Statist. Assoc.* 32, 200 (1937), 675–701.

[57] M. Friedman. 1940. A comparison of alternative tests of significance for the problem of m rankings. In *Ann. Math. Statist.* 11, 1 (1940), 86–92.

[58] F. Song, Z. Guo, and D. Mei. 2010. Feature selection using principal component analysis. In *IEEE International Conference on System Science, Engineering Design and Manufacturing Informatization* 27–30. DOI: 10.1109/ICSEM.2010.14

[59] P. K. Singh, S. Das, R. Sarkar, and M. Nasipuri. 2018. Feature selection using harmony search for script identification from handwritten document images. *J. Intell. Syst.* 27, 3 (2018), 465–488.

[60] Sk. Md. Obaidullah, C. Halder, K. C. Santosh, N. Das, and K. Roy. 2018. PHDIndic_11: Page level handwritten document image dataset of 11 official Indic scripts for script identification. In *Multim. Tools Applic.* 77, 2 (2018), 1643–1678.

[61] R. Hussain, A. Raza, I. Siddiqi, K. Khurshid, and C. Djeddi. 2015. A comprehensive survey of handwritten document benchmarks: Structure, usage and evaluation. *EURASIP J. Image Video Proces.* 5, 1 (2015) 46, 1–24.

[62] N. Esfandian, F. Razzazi, and A. Behrad. 2012. A clustering based feature selection method in spectro-temporal domain for speech recognition. *Eng. Applic. Artif. Intell.* 25, 6 (2012), 1194–1202.

[63] T. Chakraborti and A. Chatterjee. 2014. A novel binary adaptive weight GSA-based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns. *Eng. Applic. Artif. Intell.* 33, (2014), 80–90.

[64] L. Dora, S. Agrawal, R. Panda, and A. Abraham. 2017. An evolutionary single Gabor kernel-based filter approach to face recognition. *Eng. Applic. Artif. Intell.* 62, (2017), 286–301.

[65] R. Sarkar, M. Ghosh, A. Chatterjee, and S. Malakar. 2018. An advanced particle swarm optimization based feature selection method for tri-script handwritten digit recognition. In *International Conference on Computational Intelligence, Communications, and Business Analytics (CICBA).* 82–94.

[66] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili. 2018. A novel multivariate filter method for feature selection in text classification problems. *Eng. Applic. Artif. Intell.* 70 (2018), 25–37.

[67] P. K. Singh, S. Das, S. Bhowmik, R. Sarkar, and M. Nasipuri. 2016. Feature selection for Bangla handwritten word recognition using harmony search. *Int. J. Inf. Process.* 10, 4 (2016), 20–33.

[68] M. Ghosh, S. Begum, R. Sarkar, D. Chakraborty, and U. Maulik. 2019. Recursive memetic algorithm for gene selection in microarray data. In *Exp. Syst. Applic.* 116 (2019), 172–185.

[69] I. Chatterjee, M. Ghosh, P. K. Singh, R. Sarkar, and M. Nasipuri. 2019. A clustering-based feature selection framework for handwritten Indic script classification. *Exp. Syst.* 36, 6 (2019). DOI: https://doi.org/10.1111/exsy.12459

[70] D. Ciresan, U. Meier, and J. Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 3642–3649.

[71] Kwang-Eun Ko and Kwee-Bo Sim. 2018. Deep convolutional framework for abnormal behavior detection in a smart surveillance system. *Eng. Applic. Artif. Intell.* 67 (2018), 226–234,

[72] Yiqun Dong. 2019. Implementing deep learning for comprehensive aircraft icing and actuator/sensor fault detection/identification. *Eng. Applic. Artif. Intell.* 83 (2019), 28–44,

[73] Xiukun Wei, Ziming Yang, Yuxin Liu, Dehua Wei, Limin Jia, and Yujie Li. 2019. Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Eng. Applic. Artif. Intell.* 80 (2019), 66–81.

[74] Yan Zhang, Erhu Zhang, and Wanjun Chen. 2016. Deep neural network for halftone image classification based on sparse auto-encoder. *Eng. Applic. Artif. Intell.* 50 (2016), 245–255

[75] Kwang-Eun Ko and Kwee-Bo Sim. 2018. Deep convolutional framework for abnormal behavior detection in a smart surveillance system. *Eng. Applic. Artif. Intell.* 67 (2018), 226–234,

[76] S. Ukil, S. Ghosh, Sk. Md. Obaidullah, K. C. Santosh, K. Roy, and N. Das. 2018. Deep learning for word level handwritten Indic script identification. *CoRR* abs/1801.01627 (2018).

[77] S. Ukil, S. Ghosh, Sk. Md. Obaidullah, K. C. Santosh, K. Roy, and N. Das. 2019. Improved word level handwritten Indic script identification by integrating small convolutional neural networks. *Neural Comput. Applic.* 1–16. DOI: https://doi.org/10.1007/s00521-019-04111-1

Q8

[78] S. Kundu, S. Paul, P. K. Singh, R. Sarkar, and M. Nasipuri. 2020. Understanding NFC-Net: A deep learning approach to word level handwritten Indic script recognition. *Neural Comput. Applic.* 32, 12 (2020), 7879–7895. DOI: https://doi.org/10.1007/s00521-019-04235-4

[79] P. Shivakumara, Z. Yuan, D. Zhao, T. Lu, and C. L. Tan. 2015. New gradient-spatial-structural features for video script identification. *Comput. Vis. Image Underst.* 130 (2015), 35–53.

[80] Z. Malik, A. Mirza, A. Bennour, I. Siddiqi, and C. Djeddi. 2015. Video script identification using a combination of textural features. In *11th IEEE International Conference on Signal-Image Technology & Internet-based Systems (SITIS.* 61–67. DOI: 10.1109/SITIS.2015.15

[81] A. A. K. Bhunia, A. Konwer, A. K. Bhunia, A. Bhowmick, P. P. Roy, and U. Pal. 2019. Script identification in natural scene image and video frames using an attention based convolutional-LSTM network. *Pattern Recogn.* 85 (2019), 172–184.

[82] N. Sharma, S. Chanda, U. Pal, and M. Blumenstein. 2013. Word-wise script identification from video frames. In *12th International Conference on Document Analysis and Recognition.* 867–871. DOI: 10.1109/ICDAR.2013.177

[83] A. P. Gaikwad, R. R. Manza, and G. R. Manza. 2014. Automatic video scene segmentation to separate script and recognition. In *3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA).* 225–235.

[84] H. Lin, P. Yang, and F. Zhang. 2019. Review of scene text detection and recognition. *Arch. Comput. Meth. Eng.* (2019), 1–22. DOI: https://doi.org/10.1007/s11831-019-09315-1

[85] B. Shi, X. Bai, and C. Yao. 2016. Script identification in the wild via discriminative convolutional neural network. *Pattern Recogn.* 52 (2016) 448–458.

[86] B. Su and S. Lu. 2017. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recogn.* 63 (2017), 397–405.

[87] S. Tian, S. Lu, and C. Li. 2017. WeText scene text detection under weak supervision. In *IEEE International Conference on Computer Vision.* 1501–1509

[88] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan. 2014. A robust arbitrary text detection system for natural scene images. *Exp. Syst. Applic.* 41, 18 (2014), 8027–8048.

[89] X. Bai, C. Yao, and W. Liu. 2016. Strokelets: A learned multi-scale mid-level representation for scene text recognition. *IEEE Trans. Image Process.* 25 (2016), 2789–2802

[90] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: An efficient and accurate scene text detector. arXiv:1704.03155v2, 2017.

[91] Chuhui Xue, Shijian Lu, and Fangneng Zhan. 2018. Accurate scene text detection through border semantics awareness and bootstrapping. In *15th European Conference on Computer Vision (LNCS, vol. 11220).* 370–387.

[92] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. 2016. Robust scene text recognition with automatic rectification. In *IEEE Conference on Computer Vision and Pattern Recognition.* 1–9.

[93] H. Cho, M. Sung, and B. Jun. 2016. CannyText detector fast and robust scene text localization algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition.* 3566–3573.

[94] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai. 2018. Rotation-sensitive regression for oriented scene text detection. In *IEEE Conference on Computer Vision and Pattern Recognition.* 1–10.

[95] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou. 2017. Focusing attention: Towards accurate text recognition in natural images. In *IEEE International Conference on Computer Vision (ICCV).* 5086–5094.

[96] D. Dan, L. Haifeng, L. Xuelong, and C. Deng. 2018. PixelLink: Detecting scene text via instance segmentation. In *32nd AAAI Conference on Artificial Intelligence.* 6773–6780.

[97] L. Fei, K. Wang, S. Lin, K. Yang, R. Cheng, and H. Chen. 2018. Scene text detection and recognition system for visually impaired people in real world. *Proc. SPIE 10794, Target and Background Signatures IV, 107940S* (9 Oct. 2018). DOI: https://doi.org/10.1117/12.2325523

[98] Y. Tang and X. Wu. 2018. Scene text detection using superpixel-based stroke feature transform and deep learning based region classification. *IEEE Trans. Multim.* 20, 9 (2018), 2276–2288.

[99] Y. Dai, Z. Wang, X. Zhao, and S. Shao. 2018. Scene text detection based on enhanced multi-channels MSER and a fast text grouping process. *Int. J. Comput. Ling. Res.* 9 2, (2018) 47–59.

[100] A. Sain, A. K. Bhunia, Partha P. Roy, and U. Pal. 2018. Multi-oriented text detection and verification in video frames and scene images. *Neurocomputing* 275 (2018), 1531–1549.

[101] S. Dey, P. Shivakumara, R. K. Srinivas, U. Pal, T. Lu, G. H. Kumar, and C. S. Chan. 2017. Script independent approach for multi-oriented text detection in scene image. *Neurocomputing* 242 (2017), 96–112.

[102] N. Dutta, N. Chakraborty, A. F. Mollah, S. Basu, and R. Sarkar. 2019. Multi-lingual text localization from camera captured images based on foreground homogeneity analysis. *Rec. Devel. Mach. Learn. Data Anal.* 149–158. Springer, Singapore.

[103] A. Mukhopadhyay, S. Kumar, S. R. Chowdhury, N. Chakraborty, A. F. Mollah, S. Basu, and R. Sarkar. 2019. Multi-lingual scene text detection using one-class classifier. *Int. J. Comput. Vis. Image Process.* 9, 2 (2019), 48–65.

[104] M. Jajoo, N. Chakraborty, A. F. Mollah, S. Basu, and R. Sarkar. 2019. Script identification from camera-captured multi-script scene text components. *Rec. Devel. Mach. Learn. Data Anal.* (159–166).

[105] N. Chakraborty, A. Chatterjee, P. K. Singh, A. F. Mollah, and R. Sarkar. 2020. Application of daisy descriptor for language identification in the wild. In *Multimedia Tools and Applications.* Springer Publishers, 1–22. DOI: https://doi.org/10.1007/s11042-020-09728-2

[106] A. Durou, I. Aref, S. Al-Maadeed, A. Bouridane, and E. Benkhelifa. 2019. Writer identification approach based on bag of words with OBI features. *Inf. Process. Manag.* 52, 2 (2019), 354–366.

[107] I. Siddiqi and N. Vincent. 2007. Writer identification in handwritten documents. In *9th IEEE International Conference on Document Analysis and Recognition (ICDAR).* 108–112.

[108] K. Kalra and S. Rani. 2017. Writer identification from offline isolated handwritten Gurumukhi characters. In *Adv. Comput. Sci. Technol.* 10, 5 (2017), 903–914.

[109] P. Bangarimath, D. Bendigeri, and J. Pujari. 2018. Writer identification using texture features in Kannada handwritten documents. *Int. J. Eng. Res. Technol.* 6, 13 (2018), 1–5.

[110] S. Reddy, C. Andrew, U. Pal, A. Alaei, and V. Pulabaigari. 2017. Writer identification in Indic scripts: A stroke distribution based approach. In *4th Asian Conference on Pattern Recognition (ACPR).* 947–952.

[111] S. Chanda, K. Franke, U. Pal, and T. Wakabayashi. 2010. Text independent writer identification for Bengali script. In *20th IEEE International Conference on Pattern Recognition (ICPR)*. 2005–2008.

[112] C. Andrew, S. Reddy, V. Pulabaigari, and U. Pal. 2017. Text independent writer identification for Telugu script using directional filter based features. In *ICDAR-2017 Workshop on Machine Learning*. 65–70.

[113] S. Chanda, K. Franke, and U. Pal. 2012. Text independent writer identification for Oriya script. In *10th IAPR International Workshop on Document Analysis Systems*. 369–373.

[114] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. 2003. ICDAR 2003 robust reading competitions. In *7th IEEE International Conference on Document Analysis and Recognition*. 682–687.

[115] S. M. Lucas. 2005. ICDAR 2005 text locating competition results. In *8th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. 80–84.

[116] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy. 2011. ICDAR 2011 robust reading competition - Challenge 1: Reading text in born-digital images (web and email). In *11th IEEE International Conference of Document Analysis and Recognition*. 1485–1490.

[117] Asif Shahab, Faisal Shafait, and Andreas Dengel. 2011. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *11th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society. 1491–1496.

[118] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez, S. Robles, J. Mas, D. Fernandez, J. Almazan, and L. P. de las Heras. 2013. ICDAR 2013 robust reading competition. In *12th IEEE International Conference of Document Analysis and Recognition*. 1115–1124

[119] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. 2015. ICDAR 2015 competition on robust reading. In *13th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, 1156–1160.

[120] R. Gomez, B. Shi, L. Gomez, L. Neumann, A. Veit, J. Matas, S. Belongie, and D. Karatzas. 2017. ICDAR2017 robust reading challenge on COCO-Text. In *14th IEEE IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 1435–1443.

[121] M. Iwamura, N. Morimoto, K. Tainaka, D. Bazazian, L. Gomez, and D. Karatzas. 2017. ICDAR2017 robust reading challenge on omnidirectional video. In *14th IEEE IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 1448–1453.

[122] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, and W. Khlif. 2017. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT. In *14th IEEE IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 1454–1459.

[123] S. Sen, R. Sarkar, and K. Roy. 2017. An approach to stroke-based online handwritten Bangla character recognition. In *Advanced Computing and Systems for Security*. R. Chaki, K. Saeed, A. Cortesi, and N. Chaki (Eds.). Springer, 153–163.

Q12 [124] S. Sen, S. Chowdhury, M. Mitra, F. Schwenker, R. Sarkar, and K. Roy. 2018. A novel segmentation technique for online handwritten Bangla words. *Pattern Recogn. Lett.* (2018). DOI: https://doi.org/10.1016/j.patrec.2018.02.008

[125] A. K. Kinjarapu, K. C. Yelavarti, and K. P. Valurouthu. 2016. Online recognition of handwritten Telugu script characters. In *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*. 426–432.

[126] H. Swethalakshmi, C. C. Sekhar, and V. S. Chakravarthy. 2007. Spatiostructural features for recognition of online handwritten characters in Devanagari and Tamil scripts. In *International Conference on Artificial Neural Networks (ICANN) (LNCS vol. 4669)*. Springer, 230–239.

[127] N. Joshi, G. Sita, A. G. Ramakrishnan, V. Deepu, and S. Madhvanath. 2015. Machine recognition of online handwritten Devanagari characters. In *8th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, 1156–1160. DOI: http://dx.doi.org/10.1109/ICDAR.2005.156

[128] S. Kubatur, M. Sid-Ahmed and M. Ahmadi. 2012. A neural network approach to online Devanagari handwritten character recognition. In *International Conference on High Performance Computing & Simulation (HPCS)*. 209–214.

[129] L. Lajish and S. K. Kopparapu. 2015. Online handwritten Devanagari stroke recognition using extended directional features. arXiv:1501.02887v1 [cs.CV] 11.

[130] A. H. Toselli, M. Pastor, and E. Vidal. 2007. On-line handwriting recognition system for Tamil handwritten characters. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) (LNCS, vol. 4477)*. Springer, Berlin, 370–377.

[131] H. Aparna, V. Subramanian, M. Kasirajan, G. V. Prakash, V. S. Chakravarthy, and S. Madhvanath. 2004. Online handwriting recognition for Tamil. In *9th International Workshop on Frontiers in Handwriting Recognition*. 438–443.

[132] A. Sharma, R. Kumar, and R. K. Sharma. 2008. Online handwritten Gurmukhi character recognition using elastic matching. In *Congress on Image and Signal Processing*. 391–396. DOI: 10.1109/CISP.2008.297

[133] S. Singh and A. Sharma. 2019. Online handwritten Gurmukhi words recognition: An inclusive study. In *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 18, 3 (2019). DOI: https://doi.org/10.1145/3282441

[134] S. Singh, A. Sharma, and I. Chhabra. 2016. Online handwritten Gurmukhi strokes dataset based on minimal set of words. *ACM Trans. Asian Low-Resour. Lang. Info. Process.* 16, 1 (2016), 1–20.

[135] K. Prasad, I. Khan, N. R. Chanukotimath, and F. Khan. 2012. On-line handwritten character recognition system for Kannada using principal component analysis approach: For handheld devices. In *World Congress on Information and Communication Technologies*. 675–678.

[136] D. Chaithra and K. Indira. 2017. Handwritten online character recognition for single stroke Kannada characters. In *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. 548–552.

[137] R. S. Rao Kunte and R. D. S. Samuel. 2015. On-line character recognition for handwritten Kannada characters using wavelet features and neural classifier. *IETE J. Res.* 46, 5 (2015), 387–393.

[138] T. R. Indhu and V. K. Bhadran. 2012. Malayalam online handwriting recognition system: A simplified fuzzy ARTMAP approach. In *Annual IEEE India Conference (INDICON)*. 613–618. DOI: 10.1109/INDCON.2012.6420691

[139] P. Primekumar and S. M. Idiculla. 2013. On-line Malayalam handwritten character recognition using HMM and SVM. In *International Conference on Signal Processing , Image Processing & Pattern Recognition*. 322–326. DOI: 10.1109/ICSIPR.2013.6497991

[140] R. Ravindra Kumar, K. G. Sulochana, and T. R. Indhu. 2011. Online handwriting recognition for Malayalam script. In *International Conference on Information Systems for Indian Languages (Communications in Computer and Information Science, vol. 139)*. Springer, Berlin, 199–203.

[141] Q. Safdar and K. U. Khan. 2014. Online Urdu handwritten character recognition: Initial half form single stroke characters. In *12th International Conference on Frontiers of Information Technology*. 292–297. DOI: 10.1109/FIT.2014.61

[142] A. Husain, A. Sajjad, and F. Anwar. 2007. Online Urdu character recognition system. In *IAPR Conference on Machine Vision Applications (IAPR MVA)*. 98–101.

[143] S. Malik and S. A. Khan. 2005. Urdu online handwriting recognition. In *IEEE International Conference on Emerging Technologies*. 27–31.

[144] S. Naik and A. Desai. 2017. Online handwritten Gujarati character recognition using SVM, MLP, and K-NN. In *8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 1–6. DOI: 10.1109/ICCCNT.2017.8203926

[145] C. Gohel, M. M. Goswami, and V. K. Prajapati. 2015. On-line handwritten Gujarati character recognition using low level stroke. *3rd International Conference on Image Information Processing (ICIIP)*. 130–134. DOI: 10.1109/ICIIP.2015.7414753

[146] S. Joshi and Y. R. Risodkar. 2018. Deep learning based Gujarati handwritten character recognition. In *International Conference On Advances in Communication and Computing Technology (ICACCT)*. 563–566. DOI: 10.1109/ICACCT.2018.8529410

[147] A. Mukhopadhyay, P. K. Singh, R. Sarkar, and M. Nasipuri. 2020. Handwritten Indic script recognition based on the Dempster–Shafer theory of evidence. *J. Intell. Syst.* 29, 1 (2020), 264–282. DOI: https://doi.org/10.1515/jisys-2017-0431

[148] A. Mukhopadhyay, P. K. Singh, R. Sarkar, and M. Nasipuri. 2018. A study of different classifier combination approaches for handwritten Indic script recognition. *J. Imag.* 4, 39 (2018), 1–21. DOI: 10.3390/jimaging4020039

Received January 2020; revised September 2020; accepted July 2021