IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 July 2023, accepted 30 July 2023, date of publication 3 August 2023, date of current version 9 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3301564

RESEARCH ARTICLE

Efficient Assamese Word Recognition for Societal Empowerment: A Comparative Feature-Based Analysis

NAIWRITA BORAH[1], (Member, IEEE), UDAYAN BARUAH[2], MAHESH THYLORE RAMAKRISHNA[1], (Senior Member, IEEE), V. VINOTH KUMAR[3], (Member, IEEE), D. RAMYA DORAI[1], AND JONNAKUTI RAJKUMAR ANNAD[4]

[1]Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jain (Deemed-to-be University), Bengaluru 562112, India
[2]Department of Information Technology, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Gangtok 737102, India
[3]School of Information Technology and Engineering, Vellore Institute of Technology University, Vellore 632014, India
[4]Department of Electromechanical Engineering, Sawla Campus, Arba Minch University, Arba Minch 4400, Ethiopia

Corresponding author: Jonnakuti Rajkumar Annad (jonnakuti.rajkumar@amu.edu.et)

ABSTRACT The preservation and digitization of historical data are crucial for ensuring the continuity and accessibility of information over successive generations. The present study investigates the utilization of machine learning methodologies in the identification of Assamese words, focusing specifically on their distinctive visual characteristics. The main aim of this project is to improve word recognition technologies in Indic languages, specifically focusing on Assamese, in order to preserve and provide access to Assamese literature for future generations. The classification procedure entails the examination of 19 shape-related attributes through a range of machine learning algorithms, such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machine (SVM) with various kernels, K Nearest Neighbors, and Gradient Boosting. The assessment of the model involves the utilization of various metrics such as Accuracy, Precision, Kappa, F1-score, Model Build Time, and Model Run Time to evaluate the computational efficiency. Additionally, the metrics of Area under the Curve (AUC) and Receiver Operating Characteristic (ROC) are also considered in the evaluation process. Out of the four datasets analyzed, Dataset 3 exhibits the highest level of performance. It is worth noting that Gradient Boosting demonstrates the highest level of accuracy, reaching 96.03% for conventional machine learning appraoches. Logistic Regression and SVM with RBF kernel closely trail behind, achieving accuracies of 95.64% and 95.60% respectively. Furthermore, the research conducted in this study also employs multiple layers of Convolutional Neural Networks (CNN), resulting in a remarkable recognition accuracy of 97.3%. This finding demonstrates that the CNN model and the proposed feature-set are in close proximity to one another in terms of the evaluation metrics.

INDEX TERMS Feature-based approaches, comparative analysis, societal empowerment, Assamese literary works, automatic word recognition, machine learning, word image analysis, text analysis, intelligent assistive technology.

I. INTRODUCTION

The preservation and digitization of information shared by older generations are crucial for ensuring the retention and accessibility of historical data as well as the preservation of society itself. The process of digitizing books and documents is essential for preserving priceless records and cultural artifacts that the elderly share. This practice ensures their preservation, even in scenarios where the original physical copies may be susceptible to loss or damage. The digitization

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

of these materials has facilitated enhanced accessibility for scholars and researchers, thereby eliminating obstacles associated with physical inaccessibility. The implementation of sophisticated search functionalities and electronic annotations in historical archives serves to expand the audience and foster creative academic engagement. The utilization of these advanced methodologies holds significant implications for interdisciplinary research and enhances our understanding of historical events in their entirety. The interdependence between individuals and society is a crucial element, wherein the well-being of society relies on the conservation and transmission of the wisdom and knowledge possessed by its elderly constituents. Hence, it is crucial to ensure the protection and dissemination of the intellectual contributions and insights of the elderly population.

A. RESEARCH PROBLEM AND OBJECTIVES

Preserving Indic languages is crucial for ensuring precise word identification, benefiting disciplines like linguistics, literature, and history. Word recognition technology has the potential to visually identify words, providing notable benefits to the elderly population in various areas such as finance, law, and healthcare. The efficacy of its physical document handling capabilities and its potential contributions to computer vision applications have been demonstrated. The aforementioned technology facilitates data management and evaluation, supporting machine learning and natural language processing. Through the use of word recognition technology, elderly individuals can distribute digitized documents globally, preserving and sharing their intellectual contributions to advance society. The importance of recording the wisdom and experiences of older generations is emphasized by the connection between personal and communal advancement. This approach promotes harmonious living and societal advancement, particularly by empowering the elderly population. Word recognition technology can greatly benefit the elderly population in distributing digitized documents to a broader global audience. The preservation and sharing of the valuable knowledge and insights of older individuals is crucial for their personal growth and for the overall well-being of society. The wisdom and contributions of older generations are crucial for the progress and well-being of society, as there is a strong connection between the development of individuals and communities. Preserving and documenting the literary and intellectual contributions of the elderly promotes peaceful cohabitation and advances mankind, particularly by empowering and improving the welfare of the senior community.

B. CONTRIBUTION AND SIGNIFICANCE

Contribution: This study provides a comprehensive analysis of machine learning techniques for identifying Assamese script in Indic languages, focusing primarily on shape-based methodologies. This study evaluates the effectiveness of various algorithms, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machine, K Nearest Neighbors, and Gradient Boosting. This study aims to improve word recognition technologies for Assamese and other Indic languages. It will analyze nineteen shape-oriented attributes and performance metrics, such as Accuracy, Precision, Kappa, and F1-score. Segmentation techniques have been used to create a tailored dataset for literary works in the Assamese language. The dataset is used to generate a set of characteristics, which are then analyzed using the mentioned machine learning algorithms. Evaluation metrics and computational efficiency assessments are used to evaluate the accuracy and efficiency of a system.

Significance: These findings have important implications for the preservation and accessibility of cultural heritage. The study's classification of Assamese words based on their shape supports the preservation of Assamese literature and cultural heritage transmission to future generations. This study highlights the importance of digitization and language preservation efforts in protecting historical knowledge passed down by earlier generations. Advancements in word recognition technology have practical implications for language preservation, document digitization, and cultural heritage conservation. This study investigates the efficacy of machine learning algorithms in shape-based word recognition, thereby contributing to the conservation of language and cultural heritage.

C. OUTLINE OF THE PAPER

This study explores the application of machine learning techniques in shape-based word recognition for Assamese and other Indic languages. This study aims to enhance word recognition technologies while also safeguarding and promoting linguistic and cultural heritage. Section I introduces the importance of preserving historical data and the necessity of efficient word recognition systems. The article discusses the methodology (Section IV-A) and examines various machine learning algorithms used in Section IV-B, including Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines. The evaluation of classification results involves the use of performance evaluation metrics, including accuracy, precision, kappa, and F1-score, as well as computational efficiency metrics (Section IV-C). The findings indicate that there are variations in performance depending on the dataset. Specifically, Dataset 3 consistently outperforms the others in terms of multiple metrics (see Section V). The paper's Section VIII explores potential future research areas, including further investigation of shape-oriented characteristics, comparison of machine learning algorithms, and utilization of Convolutional Neural Networks.

II. LITERATURE REVIEW

An abundance of datasets are accessible for word image processing, which are commonly employed to assess the efficacy of machine learning algorithms. Table 1 showcases a range of datasets that have been utilized in the domain of document image processing, catering to diverse languages

TABLE 1. Benchmark character and word recognition databases.

S. No.	Database	Year	Language	Content	Mode (ON/OFF)	Writers
1	Sunda AMADI LontarSet [18]	2016	Balinese	100 character classes	ON	Not mentioned
2	CVL [19]	2013	English, German	Sentences	OFF	311
3	HaFT [8]	2013	Farsi	Sentences	OFF	600
4	AHTID-MW [6]	2012	Arabic	Text lines	OFF	53
5	Checks DB [20]	2012	Arabic	Check amounts	OFF	Not mentioned
6	Tamil DB [10]	2012	Tamil	Words	OFF	500
7	Devanagari DB [11]	2010	Devanagari	Digits, characters	OFF	750
8	Numerals DB [21]	2006	Bangla and Devanagari	Digits	Not mentioned	45,948
9	RIMES [12]	2006	French	Sentences	OFF	1300
10	ARABASE [7]	2005	Arabic	Sentences, words, letters	ON/OFF	400
11	IAM-OnDB [2]	2005	English	Sentences	ON	221
12	AHDB [22]	2004	English	Check amounts	OFF	Not mentioned
13	IFN Farsi [23]	2002	Farsi	Words	OFF	600
14	GRUHD [13]	2001	Greek	Text, symbols	OFF	1000
15	Firemaker [24]	2000	English	Paragraphs	OFF	250
16	IAM-HistDB [3], [4]	1999, 2002	English	Sentences	OFF	657

and document categories. The aforementioned datasets have been employed in the training and assessment of multiple word and character recognition systems. In their study, Nigam et al. (2023) conducted an extensive survey of datasets for document analysis in multiple languages, encompassing a diverse range of datasets [1]. The study carried out by them encompasses a wide range of languages and writing systems, including but not limited to the Roman, Chinese, Japanese, Arabic, and Brahmi scripts. Some datasets are limited to a particular language, while others have a broader scope that covers multiple languages. The tabulated data can be utilized to conduct a comprehensive literature review that can offer valuable insights into the accessibility and attributes of the aforementioned datasets. Numerous noteworthy datasets can be discerned. The IAM-OnDB dataset from 2005 includes images of English sentences for recognition in an online mode [2], whereas the IAM dataset from 2002 consists of handwritten English sentences in an offline context (IAM, 2002 [3]; IAM-OnDB, 2005 [4]). The MNIST dataset, established in 1998, comprises of discrete handwritten numerical digits in the English language [5]. It has been extensively utilized for the purpose of benchmarking word and character recognition. (MNIST, 1998). Additional datasets encompass AHTID-MW (2015) [6] and ARABASE (2005) for the Arabic language [7], HaFT (2013) [8] and IAUT/PHCN (2008) [9] for Farsi, as well as datasets for Tamil [10], Devanagari [11], French [12], Greek [13], Japanese [14], and Korean [15]. The NIST dataset (1995) is considered a pioneering dataset in the field of word and character recognition research [16]. It comprises isolated English digits (NIST, 1995). The CEDAR dataset, established in 1994, offers a comprehensive collection of datasets that cover English words, characters, and digits [17]. On the other hand, the PE92 dataset, established in 1993, concentrates on isolated Korean characters [15]. Both datasets have been referenced in academic literature (CEDAR, 1994 [17]; PE92, 1993 [15]). The utilization of these datasets has been instrumental in the progress of technology pertaining to the recognition of words and characters. The present literature review underscores the importance of utilizing these datasets to propel the development of word and character recognition technology across various languages and document categories. The utilization of these datasets by researchers has facilitated the training and assessment of algorithms for word and character recognition, thereby advancing the progress of precise and effective word and character recognition systems. The aforementioned datasets function as standards for assessing the efficacy of techniques for recognizing words and characters, as well as for contrasting their outcomes (see Table 1).

A. ASSAMESE LANGUAGE AND SCRIPT

The beautiful languages of Assamese, Bangla, and Oriya have fascinating origins rooted in Modern Apabhransha, which can be traced back to the ancient Magadhi-Prakrit [25]. Assamese, in particular, has a rich history filled with literary treasures like the Charyapadas, crafted by talented Buddhist scholars who shaped the language's early form [26], [27]. As time progressed, the language evolved during the pre-Sankari era with the introduction of Kavyas, bringing structure and refinement to Assamese [26]. The Assamese script, born from the ancient Brahmi script, holds a special place in India's script evolution. Fascinating rock inscriptions and copper plates from the 5th to 9th century offer glimpses into its development, comprising a collection of 42 phonetic symbols, including vowels, consonants, and diacritical marks. What's even more interesting is that this script is not only used for Assamese but also for writing other regional languages like Bodo, Karbi, and Mising [25].

The Assamese language holds noteworthy importance in the area, serving as the official language of Assam and being acknowledged as one of India's languages. The locality possesses a significant cultural legacy, having made noteworthy advancements in the domains of art, literature, and music. Comprehending the subtleties and connotations of Assamese vocabulary presents a gratifying obstacle, given the prevalence of homophones and polysemous terms. WordNet, a language database, has been developed by experts to aid

TABLE 2. Summary Of Assamese Character And Word Recognition Research.

Sl No.	Authors	Year	Algorithm	Work	Dataset Used
1	Borgohain, O., et al. [30]	2023	CNN	Recognition of Handwritten Assamese Characters	Tezpur University dataset of online handwritten Assamese characters
2	Choudhury, A., & Sarma, K. K. [31]	2023	Hybrid one-dimensional convolutional recurrent attention framework model	Trajectory-based recognition of in-air handwritten Assamese words using a hybrid classifier network	Air-written Assamese word dataset and air-written Latin words.
3	Das, R., & Singh, T. D. [32]	2022	CNN-LSTM and attention	Assamese news image caption generation using attention mechanism	Assamese news image dataset collected from various online local e-newspapers
4	Choudhury, A., & Sarma, K. K. [33]	2021	Template Matching	Offline handwritten Assamese character recognition	Not specified
5	Medhi, K., & Kalita, S. K. [34]	2020	Supervised Fuzzy Logic	Assamese Handwritten Character Recognition using Supervised Fuzzy Logic	ISI Kolkata
6	Mandal, S., et al. [35]	2018	DNN-HMM	Large vocabulary online handwritten Assamese word recognition system	Locally collected Assamese word database
7	Baruah, U., & Hazarika, S. M. [36]	2015	SVM	A Dataset of Online Handwritten Assamese Characters	Author-created dataset

in the exploration of the intricacies of Assamese language by providing insights into word attributes [28]. An Assamese WordNet with a straightforward API is the result of linguists and computer scientists working together. Sarma's study (2013) emphasizes the significance of word classification in Natural language processing (NLP) and the intricacies of comprehending contextual meaning [29]. The scholars have proposed a novel method for classifying Assamese lexemes according to their distinctive inflectional features, thereby augmenting Part-of-Speech (POS) tagging and other linguistic analysis techniques. This innovative methodology exhibits significant potential and may be expanded to other captivating inflectional languages spoken across India.

B. RELATED WORKS ON ASSAMESE WORD RECOGNITION

The recognition of Assamese words has emerged as a prominent research domain in recent times, with numerous studies being conducted in this area. The insufficiency of a suitable dataset poses a significant obstacle in the development of a system that can accurately recognise Assamese words.

The identification of Assamese characters and terms has been the subject of extensive research, as shown in Table 2. By addressing issues with various fonts and intricate character structures, these studies collectively advance the field of Assamese Document analysis.

For the purpose of identifying handwritten Assamese characters, Borgohain et al. present the CNN approach [30]. By utilizing a variety of hybrid methodologies, Choudhury and Sarma have advanced the field of in-air handwritten word and character recognition [31]. Das and Singh concentrate on using CNN-LSTM and attention mechanisms to produce image descriptions for Assamese news [32]. For the offline recognition of handwritten characters, Choudhury and Sarma use template matching [33]. To identify characters, Medhi and Kalita used supervised fuzzy logic [34]. For the purpose of recognizing online handwritten Assamese words, Mandal et al. suggest a deep neural network-hidden Markov model (DNN-HMM) architecture [35]. Using SVM, Baruah and Hazarika created an online dataset of handwritten Assamese characters [36].

III. DATASET USED

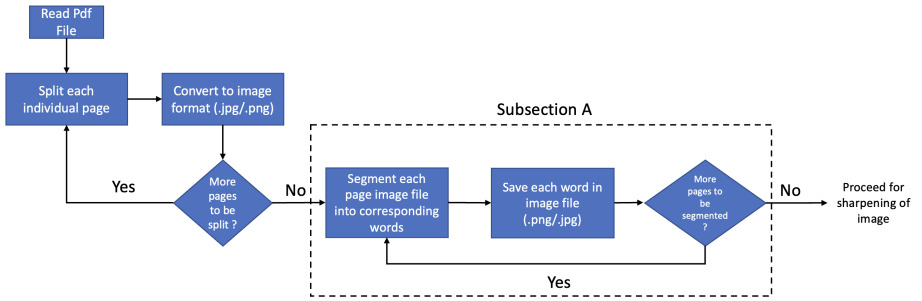
The dataset utilized in the experiments was manually curated. The images were produced using publicly accessible online materials pertaining to the literary periods of Jonaki and Shankari in Assamese literature.

A. PRE-PROCESSING

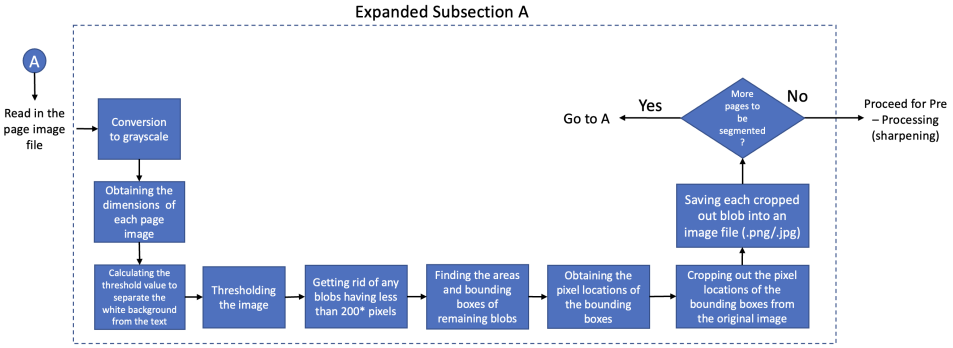
A considerable number of document images exhibit diverse types of degradation, such as inadequate contrast and noise. Pre-processing is a necessary step to ensure appropriate feature extraction. In the event that the input is a colored image, the initial step involves the conversion of said image into a grayscale format. Mean filter masks are commonly employed in the process of noise reduction. The query word or character image is subjected to binarization through the Otsu thresholding [37], resulting in the foreground, or text region, being assigned a value of 1, while the background, or non-text region, is assigned a value of 0.

B. WORD IMAGE SEGMENTATION

Upon submission of the Portable Document Format (PDF) to the system, each page undergoes conversion to a page-level image. The page image is then segmented to extract word-level images, which are saved as *"BookName_PageNumber_WordNumber.png"*. Figure 1 depicts the algorithm used to prepare datasets. Figure 1a illustrates the procedural sequence of word image segmentation, providing a thorough summary of the procedure and its phases. The segmentation steps are represented visually to facilitate comprehension. Figure 1b is a workflow diagram that provides an in-depth analysis of Subsection A of the word image segmentation process and a comprehensive view of the algorithm. To improve image processing,



(a) Segmentation Workflow



(b) Expanded Subsection A

(c) Sample Images

FIGURE 1. **Word image segmentation workflow and corresponding outcome.**

extracted word-level images undergo standard preprocessing techniques such as noise removal and sharpening. The normalization of Assamese word images to a size of 256 by 64 is performed, followed by thinning, taking into account their horizontal writing orientation. Thinning is a method that reduces the width of text from multiple pixels to a single pixel, thereby decreasing the amount of data necessary for the depiction or retention of a word.

By using this segmentation method, four datasets were generated from five books, namely

- *Burhi_Aair_Xadhu*
- *Bezbaruahr Rasanawali (Vol 2)*
- *Kirttana_and_Ghosha*
- *Gauburha*
- *Jilikoni*

The nomenclature used to identify each image is as *"Book-Name_PageNumber_WordNumber.png"*. Each book corresponds to a particular class for easy classification and identification. Figure 1c gives a sample of the images belonging to the book (or class) ‘‘Jilikoni’’. Each dataset contains a nearly equivalent number of images extracted from each book. Table 3 showcases four distinct datasets, each comprising of instance counts pertaining to various class labels. Each individual dataset is associated with a distinct collection of literary works. The datasets range from 2,417 to 8,561 instances in total. The class labels correspond to the titles of the books, and the instance count represents the

TABLE 3. Datasets With Count Of Images For Each Class Label.

Dataset 1			Dataset 2		
No.	**Class Label**	**Instance Count**	**No.**	**Class Label**	**Instance Count**
1	Burhi_aair_xadhu	1250	1	Burhi_aair_xadhu	536
2	Bezbaruahr Rasanawali (Vol 2)	1275	2	Bezbaruahr Rasanawali (Vol 2)	354
3	Kirttana_and_Ghosha	1252	3	Kirttana_and_Ghosha	496
4	Gauburha	1298	4	Gauburha	416
5	Jilikoni	1270	5	Jilikoni	615
Total		**6345**	**Total**		**2417**
Dataset 3			**Dataset 4**		
1	Burhi_aair_xadhu	1703	1	Burhi_aair_xadhu	1095
2	Bezbaruahr Rasanawali (Vol 2)	1700	2	Bezbaruahr Rasanawali (Vol 2)	1091
3	Kirttana_and_Ghosha	1726	3	Kirttana_and_Ghosha	1095
4	Gauburha	1716	4	Gauburha	1080
5	Jilikoni	1716	5	Jilikoni	1090
Total		**8561**	**Total**		**5451**

number of images for each class. Dataset 1 contains 6,345 instances, while Dataset 2 contains 2,417 instances, with varying frequencies for each class label. There are 8,561 images in Dataset 3, and 5,451 instances in Dataset 4, each with a unique class label count in Dataset 4. The table 3 provides an exhaustive overview of the datasets, including class labels and instance numbers. The datasets are mutually exclusive, resulting in 22,744 unique images in total.

IV. METHODOLOGY

Ideally, in Content-Based Image Retrieval (CBIR) approaches, color, texture, and shape characteristics play a significant role in the recognition and identification phases. In word recognition experiments, however, color and texture play no significant role. Therefore, in this study, shape-based approaches are emphasized. Section IV-A provides an elaborate explanation of the features that have been included in the study. A comprehensive set of 19 features has been incorporated across the 4 datasets, as outlined in Table 4. These 19 features collectively contribute to a total of 1523 shape-based features. These features were subsequently employed in various classification tasks using well-established classification techniques for evaluation and analysis. Section IV-B gives a detailed encounter with this.

A. FEATURES IMPLEMENTED

Upon completion of the preprocessing stage, the process of feature extraction assumes a critical role in the system of recognizing handwritten words. The aim of this study is to measure the prominent shape attributes of characters or words with the objective of optimizing the accuracy of recognition. The significance of the extracted features' quality in determining the recognition system's overall performance has been mentioned in [38]. Nonetheless, the process of extracting pertinent features from indic words can prove to be a formidable undertaking, owing to the intrinsic variability and coarseness that frequently characterizes such script. Hence, it is crucial to extract solely those characteristics that efficiently differentiate among diverse word categories within a particular recognition framework, particularly in applications such as word recognition. Table 4 gives an account of the features implemented for this study.

1) HU MOMENTS

Hu moments are a set of seven mathematical descriptors derived from the central moments of an image. Ming-Kuei Hu introduced them in 1962 as a robust technique for describing the shape and texture of objects in an image [39]. Hu moments capture various statistical moments of an object's shape and provide a compact, translation, rotation, and scaling-invariant representation.The Hu moments quantify the distribution of pixel intensities in an image in order to capture crucial shape characteristics, such as symmetry, angularity, and curvature. The invariance of Hu moments to translation, rotation, and scaling makes them robust for object recognition. Equations 1 to 7, as shown at the bottom of the next page, describe the seven Hu moments extracted from each word image.

It is one of the comparable feature descriptor that can be used in shape based image processing tasks [40].

2) EDGE HISTOGRAM DETECTION

The Edge Histogram Descriptor (EHD) is a method employed for the extraction of features for image recognition [41]. The methodology entails the utilization of edge detection algorithms to capture pertinent edge information. Subsequently, the orientations and strengths of the edges are quantified into bins, and histograms are computed. The acquired descriptor characterizes the edge distribution present in an image, thereby facilitating the scientific process of image recognition. It gives a total of 80 features. The technique of EHD is utilized to capture the spatial arrangement of edges. The aforementioned system captures edges that have been classified into five distinct categories. The five types of orientations that can be observed in a pattern or structure are as follows:

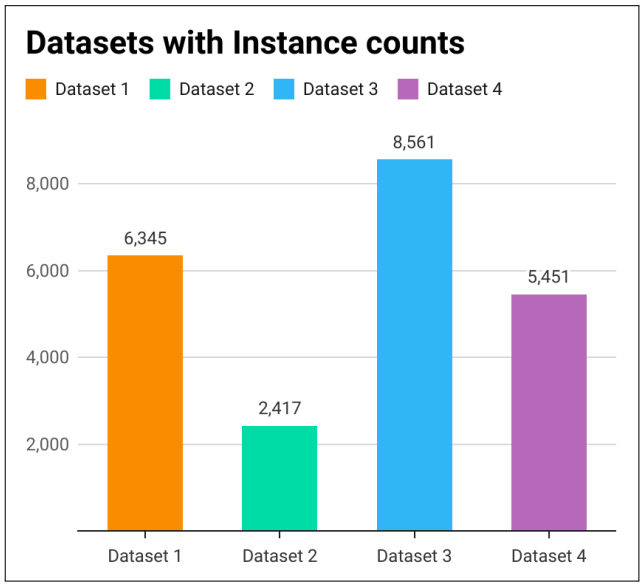


FIGURE 2. Datasets with instance counts.

1) Vertical: Edge orientations that are primarily vertical, perpendicular to the horizontal axis.
2) Horizontal: Edge orientations that are primarily horizontal, parallel to the horizontal axis.
3) Diagonal at 45 degrees: Edge orientations that form a 45-degree angle with the horizontal axis.
4) Diagonal at 135 degrees: Edge orientations that form a 135-degree angle with the horizontal axis.
5) Isotropic: Refers to a lack of orientation specificity, indicating edges that are not predominantly aligned with any specific direction.

The algorithm for the same is mentioned below:

Step 1: Image partitioning The process of partitioning an image involves dividing the input image into non-overlapping blocks of size 4×4, resulting in a total of 16 blocks. Subsequent to the extraction process, each block is partitioned into smaller 2×2 blocks to effectively capture the local edge orientation. In cases where the dimensions of an image are not evenly divisible by 4, the image is adjusted in size to ensure that both M/4 and N/4 are whole numbers.

Step 2: Capturing the local orientation of edges The process of capturing local edge orientation involves initializing a five-point bin for each block ($M/4 \times N/4$). The bin's vector components, denoted as V, H, $D45$, $D135$, and NOE, are all equal to zero, where:

- The orientation of the vertical edge is denoted by the symbol "V".
- The letter "H" denotes the orientation of a horizontal edge.
- The term "D45" refers to the diagonal edge that is oriented at a 45-degree angle.
- The term "D135" refers to the diagonal edge with an orientation of 135 degrees.
- The term NOE refers to the orientation of non-edge (isotropic) regions.

Step 3: Capturing the edge orientation from a 2×2 block. Each operator applied on a 2×2 sub-block is captured by the following 2×2 operators:

$$EO_{\text{type}} = \left| \sum a_k \cdot d_k \right| \tag{8}$$

where:

$$[a_k] = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix} \tag{9}$$

represents the 2×2 sub-block, and

$$[d_k] = \begin{bmatrix} d_0 & d_1 \\ d_2 & d_3 \end{bmatrix} \tag{10}$$

represents the edge detector. Using this method, five values are obtained: EO_v, EO_h, EO_{d45}, EO_{d135}, and EO_{noe}.

After this, the dominant edge orientation is determined by finding the maximum of these five values and comparing it to the predefined threshold (T):

$$EO_{\text{dominant}} = \max(EO_v, EO_h, EO_{d45}, EO_{d135}) > T \tag{11}$$

The EO_{dominant} will be equal to any of the five orientations with the maximum value. The count of the corresponding bin point is increased by 1 because they were all initialized to 0.

$$H_1 = \eta_{20} + \eta_{02}, \tag{1}$$

$$H_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \tag{2}$$

$$H_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \tag{3}$$

$$H_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \tag{4}$$

$$H_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \tag{5}$$

$$H_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), \tag{6}$$

$$H_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \tag{7}$$

FIGURE 3. Image partitioning for EHD.

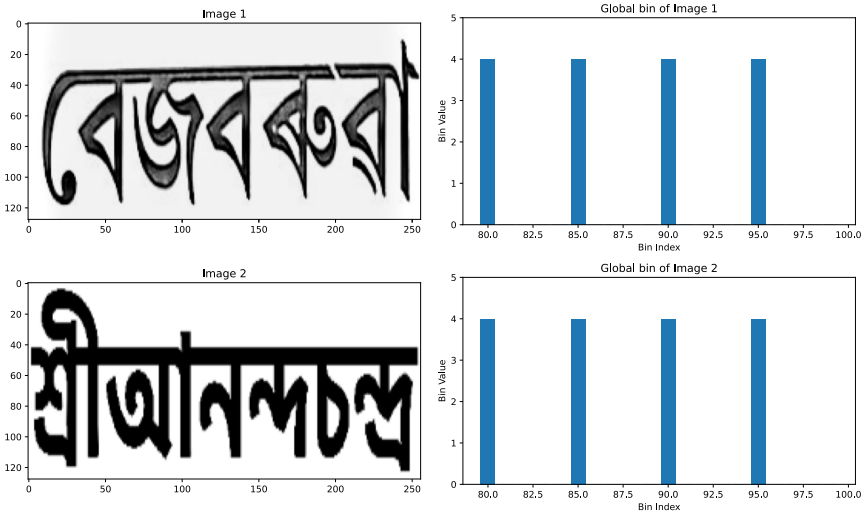


FIGURE 4. Word image and EHD global bins.

This process is repeated for all 2×2 sub-blocks in one image block.

For one image block, the complete bin is obtained as follows:

$$Bin[1] = [b_0, b_1, b_2, b_3, b_4] \tag{12}$$

The above operation is repeated for all 16 image blocks to obtain 16 bins. Then, the bins for all 16 image blocks are arranged as follows:

$$[AllBins] = (b_0 b_1 b_2 b_3 b_4\ b_5 b_6 b_7 b_8 b_9 \ldots\ldots b_{75} b_{76} b_{77} b_{78} b_{79}) \tag{13}$$

The global bin is found by taking the mean of the matrix *AllBins*:

$$GlobalBin = \text{mean}(AllBins) \tag{14}$$

The concatenation of the global bin, denoting the average bin, and the derived bins yields an EHD vector with a dimension of 85.

$$EHD = [Bin[1], Bin[2], \ldots, Bin[16], GlobalBin] \tag{15}$$

Additionally, the following equations are used in the process:

$$EO_v = \left| \sum a_k \cdot d_{kv} \right| \tag{16}$$

$$EO_h = \left| \sum a_k \cdot d_{kh} \right| \tag{17}$$

$$EO_{d45} = \left| \sum a_k \cdot d_{kd45} \right| \tag{18}$$

$$EO_{d135} = \left| \sum a_k \cdot d_{kd135} \right| \tag{19}$$

Figure 4 gives a visual representation of some of the global bins for 2 images.

3) SHAPE INDEX

The Shape Index (SI) is a scalar metric that quantifies the degree of curvature in a given locality. It is obtained by evaluating the eigenvalues of the Hessian matrix, as originally proposed by Koenderink and van Doorn [42]. This method has the capability to identify structures by their observable local morphology. The numerical scale known as SI encompasses a range from -1 to 1 and serves to indicate diverse shapes. The SI at a given point in an image is typically computed by utilizing the Hessian matrix's eigenvalues. The matrix represents the localized curvature characteristics of the

image. The Hessian matrix is a square matrix consisting of second-order partial derivatives.

$$\text{ShapeIndex} = \frac{2\kappa_1}{\kappa_1 + \kappa_2} \tag{20}$$

where:

κ_1 : Principal curvature in the first direction,

κ_2 : Principal curvature in the second direction.

The mathematical expressions for the principal curvatures are as follows:

$$\kappa_1 = \frac{L_1 + \sqrt{L_1^2 - 4M_1}}{2} \tag{21}$$

$$\kappa_2 = \frac{L_2 + \sqrt{L_2^2 - 4M_2}}{2} \tag{22}$$

Here, L_1 and L_2 represent the sum of the eigenvalues of the Hessian matrix, which characterizes the local curvature at a given point. M_1 and M_2 denote the determinant of the Hessian matrix at the specified point.

The principal curvatures provide valuable insights into the curvatures along two perpendicular directions, aiding in the understanding of shape characteristics in the fields of computer vision and image processing.

4) SPATIAL PYRAMID OF LOCAL BINARY PATTERNS

In computer vision and image processing, the Spatial Pyramid of Local Binary Patterns (SPLBP) is a prevalent computational technique [43]. It is employed for shape-oriented analysis and feature extraction by computing binary patterns within image patches at different scales and positions. By comparing pixel intensities, binary patterns are converted into codes, and histograms capture pattern distribution for texture and shape analysis. SPLBP employs a pyramidal structure and region overlaps for exhaustive shape analysis. P_r and P_{ru} represent sampling points with circular symmetry on radius r, whereas $I_{w\times w}$ represents the image patch size. SPLBP computes LBP by considering sampling points that are symmetrically positioned on the radius surrounding the central pixel.

The LBP operator denoted as LBP p is defined as follows:

$$LBP_p = \begin{cases} 1, & \text{if } I(p) \geq I_c \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

where $I(p)$ represents the intensity value at a given sampling point p, and I_c represents the intensity value of the center pixel. Combining LBP histograms calculated from circularly symmetric sampling points p yields the SPLBP feature. The expression denotes the SPLBP sequence of P_{ru} elements denoted as H_i $(1 \leq i \leq P_{ru})$.

H_p represents the histogram of LBP values for a sampling point with circular symmetry.

The SPLBP operator is widely employed at multiple scales, and its concatenation produces a spatial pyramid representation. This method captures diverse local patterns present at various scales and locations within the image patch.

5) CONTOUR LENGTH

Contour Length is a metric that quantifies the perimeter of an image entity, providing insight into its size, intricacy, and spatial scope. The application of this technique is widespread in object recognition, shape analysis, and image segmentation. The length of an object can be determined by tracing its perimeter and calculating the total path length along the contour. Perimeter computation techniques encompass the addition of consecutive boundary point distances or the utilization of algorithms, such as the Freeman chain code [44] or the Douglas-Peucker algorithm [45].

The length of the contour is determined by adding the Euclidean distances between successive points on the curve. It is the total length or perimeter of a curve or border. This expression represents the calculation:

$$CL = \sum_{i=1}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{24}$$

Here, n represents the total number of contour points, and (x_i, y_i) represents the coordinates of the i-th contour point. The formula computes the Euclidean distance between each consecutive pair of points and adds them to determine the total length of the contour. The computation of contour length provides a quantitative evaluation of the entity's boundary or configuration, which has applications in fields such as object recognition, shape evaluation, and boundary analysis.

6) CONVEX HULL AREA AND CONVEX HULL PERIMETER

The Convex Hull is a minimal convex polygon that encloses and represents the extremity of a set of 2D points. The Convex Hull Area (CHA) measures the object's spatial extent. Several algorithms compute the area of the Convex Hull, including Gift Wrapping, Graham's Scan, and Quick-Hull [46], [47], [48].

The Convex Hull Perimeter (CHP) is a measurement of the perimeter of this complex geometric shape that reveals its complexity. Adding the Euclidean distances between adjacent Convex Hull vertices is required. The circumference of the Convex Hull is represented by P_{CH}.

A_{CH} and CH denote, respectively, the Convex Hull Area and Convex Hull.

The Convex Hull Area, denoted by A_{CH}, is calculable using the vertex coordinates in the following equation:

$$A_{CH} = \frac{1}{2} \sum_{i=1}^{n} (x_i y_{i+1} - x_{i+1} y_i) \tag{25}$$

where n is the number of vertices and (x_i, y_i) represents the coordinates of the i-th vertex. To determine the area of each triangle that forms the Convex Hull, it is necessary to evaluate the cross-product of consecutive vertices' coordinates. The total area of the triangles is divided by 2.

The CHP, denoted by P_{CH}, can be calculated as follows:

$$P_{CH} = \sum_{i=1}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{26}$$

It involves adding the Euclidean distances between adjacent Convex Hull vertices. The range of the index i is between 1 and n, where n is the total number of vertices. (x_i, y_i) and (x_{i-1}, y_{i-1}) represent the i-th and $(i-1)$-th vertices, respectively.

7) KURTOSIS

Kurtosis is a metric used in statistics to quantify the shape and distribution characteristics of a dataset [49]. Positive kurtosis denotes larger peaks and heavier tails, whereas negative kurtosis denotes smaller peaks and lighter tails.

The sample kurtosis, denoted by K, is calculated using the formula:

$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} \tag{27}$$

where x_i is the i-th observation, $\bar{x}$ is the sample mean, and n is the total count of observations.

Kurtosis values provide information about the distribution's shape.

- $K > 0$: Indicates a leptokurtic distribution with more pronounced peaks and longer tails.
- Similar to the normal distribution, $K = 0$ represents the mesokurtic distribution.
- $K < 0$: Indicates a platykurtic distribution with less pronounced peaks and shorter tails.

When interpreting kurtosis values, it is important to consider the context and the specific attributes of the dataset.

8) PYRAMID OF HISTOGRAMS OF ORIENTED GRADIENTS

The Pyramid of Histograms of Oriented Gradients (PHOG) algorithm is extensively employed for feature extraction in computer vision and image processing [50]. It captures shape and texture information by utilizing gradients at different scales and levels.

The PHOG algorithm can be summarized as follows:

1) **Computing Gradients:** Calculate the magnitude $M(x, y)$ and orientation $\theta(x, y)$ of gradients for each pixel in the image using gradient operators like Sobel or Scharr.
2) **Histogram Production:** Divide the visual representation into cells and compute a histogram of gradient orientations for each cell. The orientations are discretized into a predetermined number of bins (e.g., nine bins covering 0 to 180 degrees). Denote the histogram value for bin i in cell c as $H_c(i)$.
3) **Standardization of Blocks:** Divide the image into blocks comprising multiple cells to capture spatial relationships. Normalize the histograms within each block to handle illumination variations and enhance discriminative properties.
4) **Pyramid Architecture:** Generate a hierarchical structure of histograms by iteratively dividing the image into finer regions or scales. Construct the pyramid by aggregating histograms from adjacent cells or blocks at different scales. Pooling techniques like spatial pooling and max pooling can be used.
5) **Attribute Extraction:** Extract features from the histogram pyramid. These features can serve as input for various machine learning algorithms for tasks such as object detection, classification, and recognition.

Calculating gradients, generating histograms, performing pooling operations, and extracting features from the histogram pyramid are all components of the PHOG algorithm. It permits the calculation of histogram values for particular cells.

The representation of the histogram calculation for a cell c is as follows:

$$H_c(i) = \sum_{(x,y)\in c} \delta(\theta(x, y) - i) \cdot M(x, y) \tag{28}$$

Here, $\delta(\cdot)$ is the Kronecker delta function, $\theta(x, y)$ is the orientation of the gradient at pixel (x, y), and $M(x, y)$ is the magnitude of the gradient.

Methods such as L2 normalization and power normalization can be utilized to perform block normalization. L2 normalization can be expressed as follows for a block B and histogram value $H_B(i)$ in bin i:

$$H_B(i) = \frac{H_B(i)}{\sqrt{\sum_i H_B(i)^2 + \epsilon}} \tag{29}$$

The pooling operation combines the histogram values of adjacent cells or blocks. The max pooling operation for a pooled histogram value $P(i)$ is represented as:

$$P(i) = \max_j H(j) \tag{30}$$

Here, j represents the indexes of the cells or blocks that have been pooled.

PHOG is an efficient algorithm for feature extraction that is widely employed for object recognition and detection. It outperforms numerous computer vision benchmarks by providing robust and discriminative shape and texture information at multiple scales and levels of detail.

9) GEOMETRIC FEATURES

This section will examine prevalent geometric characteristics utilized for the analysis and depiction of shapes.

1) **Rectangularity**: Measures how much an object looks like a rectangle by comparing the area of the object to the area of the smallest bounding rectangle.

$$\text{Rectangularity} = \frac{\text{Area (Object)}}{\text{Area (Bounding Rectangle)}} \tag{31}$$

2) **Volume**: Measures how much space a 3D object takes up.

3) **Compactness**: Uses the square root of the area to figure out how well an object's perimeter surrounds its area.

$$\text{Compactness} = \frac{\text{Length}}{\sqrt{\text{Area}}} \tag{32}$$

4) **Major Axis Length (MaAL)**: This is the length of an object's longest dimension. It can be found using methods like principal component analysis or ellipse fitting.

$$\text{MaAL} = \max\left(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}\right) \tag{33}$$

5) **Minor Axis Length (MiAL)**: This is the shortest distance between two points on an object. It is calculated the same way as the major axis length.

$$\text{MiAL} = \min\left(\sqrt{(x_i - x)^2 + (y_i - y)^2}\right) \tag{34}$$

6) **Number of Corners**: Uses the Harris Corner Detector or the Shi-Tomasi corner detection algorithm to find corners.

7) **Number of Holes**: Uses contour detection and topological analysis to find out how many holes an object has.

$$\text{NumHoles} = \text{Count (Contour hierarchy above 0)} \tag{35}$$

8) **Perimeter**: The total length of an object's border. This is often used to find an object's shape.

$$\text{Length (closed contours)} = \text{Perimeter} \tag{36}$$

9) **Skewness**: A way to measure how uneven a shape or distribution is.

$$\text{Skewness} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{n \cdot \sigma^3} \tag{37}$$

10) **Solidity**: Figures out the ratio between the area of an object and the area of its convex hull, which shows how much space it takes up.

$$\text{Solidity} = \frac{\text{Area of the object}}{\text{Area of the convex hull}} \tag{38}$$

These features give us important information about the characteristics and properties of shapes, which helps us analyze and recognize them.

B. CLASSIFICATION ALGORITHMS

Following the feature extraction stage, the classification process is employed to determine the classification of an unfamiliar term by utilizing the extracted feature vectors as a basis for decision-making. In order to achieve this objective, it is necessary to train the classifiers using a designated set of training data, which enables them to correlate the extracted features with pre-established rules. This study employs multiple classifiers, namely Logistic Regression (LR), Decision Trees (DT), Random Forest (RF), Support Vector Machine (SVM) with diverse kernels, K Nearest Neighbors (KNN), and Gradient Boosting (GB). The classifiers assume a pivotal function in the classification procedure, and their concise explications are presented in the subsequent subsections.

TABLE 4. Summary Of Feature Implemented For Assamese Word Recognition.

Feature Name	Count	Description
SPLBP	756	Features describing spatial layout: position, orientation, scale.
PHOG	630	Pyramid of Histograms of Oriented Gradients
EHD	80	Edge Histogram descriptors for shape analysis.
ShapeIndex	36	Scalar value characterizing local shape based on curvature.
HuMoments	7	Moments computed from central moments for shape description.
Contlength	1	Length of object contour or boundary.
CHA	1	Area of smallest convex polygon containing the object.
CHP	1	Perimeter of smallest convex polygon containing the object.
Kurtosis	1	Measure of distribution's "peakedness" or "flatness".
Rectangularity	1	Ratio of object area to minimum bounding rectangle area.
Volume	1	Volume or space occupied by the object.
Compactness	1	Measure of object's packing density.
MaAL	1	Length of object's major axis.
MiAL	1	Length of object's minor axis.
Num_corners	1	Number of corners or vertices in the object's contour.
Num_holes	1	Number of holes or voids in the object.
Perimeter	1	Length of the object's contour.
Skewness	1	Measure of object's distribution asymmetry.
Solidity	1	Ratio of object's area to its convex hull area.
Total	1523	

1) LOGISTIC REGRESSION

Logistic Regression (LR) is a popular statistical model for classification tasks, particularly in shape-based analysis [51]. LR extends the linear regression model by estimating the probability of a binary outcome using a logistic function [52]. This method uses shape-based characteristics to make predictions. LR's applications in shape recognition, shape classification, and object detection are extensive. The model parameters are estimated using maximum likelihood estimation and likelihood function optimization. By assigning weights to shape-based features, LR generates results that are easily interpretable and permit an understanding of the relative importance of various shape characteristics. Numerous studies have demonstrated the effectiveness of LR for shape-based analysis.

Following is the mathematical expression of the LR model:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)}} \tag{39}$$

where:

- Y is the binary dependent variable,

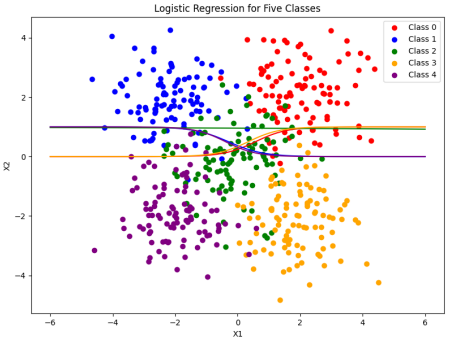


FIGURE 5. Example of LR with 5 sample classes.

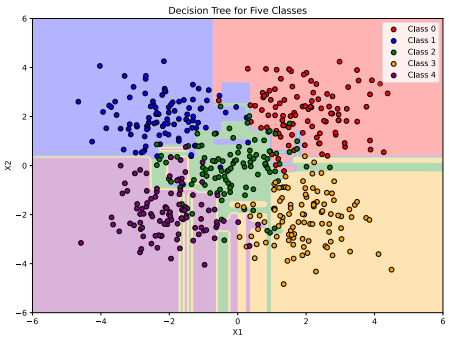


FIGURE 6. Example of DT with 5 sample classes.

- X is a vector of independent variables,
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ are the coefficients or parameters of the model,
- $X_1, X_2, \ldots, X_p$ are the corresponding independent variables.

It computes the conditional probability of the dependent variable Y being equivalent to 1, given the values of the independent variables $X_1, X_2, \ldots, X_p$.

2) DECISION TREE

Decision Trees (DT) are supervised non-parametric machine learning models used to detect word images based on their shape [53]. Analysis of shape properties is required for the classification and recognition of word images. In DT construction, shape-related characteristics extracted from word images are utilized [54]. Internal nodes represent distinctive shape characteristics, whereas branches represent decision rules based on these characteristics. Class labels or predictions for word image shapes are stored in terminal nodes. Using shape attributes, shape-based word image detection traverses the decision tree from root to leaf node in order to classify or identify the textual representation based on its shape. Despite variations in shape, the intrinsic shape attributes of DTs permit accurate classification of word images. DTs provide valuable insights into shape patterns, thereby facilitating the creation of trustworthy algorithms for CBIR and document processing. The formula for the splitting criterion varies, with the Gini index commonly used to quantify impurities.

$$\text{Gini} = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{40}$$

models measure impurity with the Gini Index and entropy. Both entropy and the Gini Index are used to reduce impurities from the root to leaf nodes in the decision tree:

$$\text{Entropy} = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{41}$$

where C is the number of classes and p_i is the probability of an instance belonging to class i. Splitting at each node aims to decrease impurity or increase information gain, resulting in improved categorization or regression. Decision Trees are simple models that can handle both numerical and categorical features. However, they are prone to overfitting and can be sensitive to minor variations in the training data. Hierarchical decision trees provide insights into decision-making and the importance of features.

3) RANDOM FOREST

Random Forest (RF) is an algorithm that utilizes bagging and stochastic feature selection to integrate multiple tree classifiers [55]. The tree classifier ensemble, denoted by $h(x, \Theta_k), k = 1 \ldots K$, is independent. Each tree's characteristics are determined stochastically.

Individual trees are generated by randomly selecting T_k observations from N total observations as a training sample. Recursive partitioning minimizes the size of the tree. Each partition takes into account a random subset of m features from F. The top subsets include divisions. Unpruned groupings of trees exhibit robust growth. The classification of an input x requires the aggregation of tree predictions, often by majority vote. The weighting of predictions may take performance into account, among other factors. 33% of training instances are excluded from Bootstrap samples during tree construction. Out-of-bag (OOB) data estimates error in the training dataset and evaluates the significance of features. RF integrates tree classifiers developed independently. Common practices include random feature selection and majority voting. Bootstrap samples and OOB data are used to estimate classifier performance and feature importance [54].

4) SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a versatile classification and regression algorithm, particularly for complex and high-dimensional data. It seeks to identify an optimal hyperplane that maximally separates data points belonging to different classes while maximizing the margin [56].

SVM employs distinct kernels to handle nonlinear data relationships. Three common kernels include:

1) Linear Kernel: Assumes linear separability and uses the inner product of data points to determine the decision

boundary:

$$K_L(x_i, x_j) = x_i^T x_j \tag{42}$$

2) Polynomial Kernel: Maps data into a higher-dimensional space using polynomials to represent complex relationships:

$$K_P(x_i, x_j) = (x_i^T x_j + c)^d \tag{43}$$

with c being a constant and d representing the degree.

3) Radial Basis Function (RBF) Kernel: Handles extremely nonlinear relationships by transforming data into an infinite-dimensional space:

$$K_R(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{44}$$

, where σ controls the width of the kernel.

SVM with multiple kernels offers classification and regression flexibility. The selection of the kernel depends on the characteristics and relationships of the data variables. SVM with shape-based features accurately classifies or regresses in word image processing by choosing a kernel that captures significant patterns and structures in the data [57].

5) K NEAREST NEIGHBORS

In shape-based Content-Based Image Retrieval (CBIR) systems, the K Nearest Neighbors (KNN) algorithm [58], [59] is widely employed [54], [60], [61]. It seeks to identify the K images from a database that are most similar to the query image based on their shape characteristics. Common shape descriptors, such as Hu moments or Fourier descriptors, are used to represent the shape's characteristics [62]. Using the following formula, the KNN algorithm computes the Euclidean distance ($dist$) between the query image (q) and each database image (d):

$$dist(q, d) = \sqrt{\sum_{i=1}^{n}(q_i - d_i)^2} \tag{45}$$

Here, q_i and d_i represent the i-th feature value of the shapes in the query image and the database image, respectively. The K-nearest neighbors (KNN) algorithm computes the distances between the query image and all database images to determine the K closest matches.

Once the K nearest neighbors have been identified, the CBIR system can display these images as potential matches to the user. Utilizing KNN in shape-based CBIR to retrieve images with similar shapes has proven effective, with applications in object recognition, medical imaging, and pattern recognition.

6) GRADIENT BOOSTING

Gradient Boosting (GB) is a versatile machine learning technique that constructs a robust predictive model by iteratively combining weak learners, typically decision trees. It demonstrates proficiency in shape identification, textual content extraction, and document categorization [63]. GB optimizes a loss function through iterative minimization of errors between predicted and actual values using gradient descent optimization. Due to its adaptability and resilience, it is highly suitable for shape-oriented textual image analysis tasks. GB is represented as:

$$F_M(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{46}$$

Here, $F_M(x)$ represents the model, M is the number of iterations, γ_m is the learning rate for the m-th iteration, and $h_m(x)$ is the learner at the m-th iteration. GB gradually integrates additional base learners into the model by adapting them to the negative gradient of the loss function with respect to the current model's predictions.

7) CONVOLUTIONAL NEURAL NETWORK

The convolutional neural network (CNN) model proposed in this study has been specifically developed for the purpose of conducting image classification tasks as CNN has seen many applications for scripts related tasks [64]. The model is designed to handle grayscale images with a resolution of 128 × 128 pixels. The model's architecture is composed of multiple layers that aim to gradually extract higher-level features from the input images. This process facilitates the learning of meaningful representations, ultimately leading to accurate classification.

The model's initial layers consist of four convolutional layers with progressively larger filter sizes: 64, 128, 256, and 512 filters, respectively. After each convolutional layer, a subsequent max-pooling layer is employed in order to perform downsampling on the feature maps and effectively decrease the spatial dimensions. The convolutional layers employ the Rectified Linear Unit (ReLU) as the activation function, which serves to introduce non-linearity and facilitate the detection of intricate patterns within the data. Figure 7 gives a representation of the model.

Following a sequence of convolutional and pooling layers, a Flatten layer is utilized to convert the three-dimensional feature maps into a one-dimensional vector. The process of flattening facilitates the seamless transition from convolutional layers to fully connected dense layers.

The subsequent layers are comprised of two dense (fully connected) layers. The initial dense layer consists of 400 neurons, which is subsequently followed by a dropout layer incorporating a dropout rate of 0.5. This dropout layer is implemented to mitigate the risk of overfitting during the training process. The dropout technique involves stochastically deactivating a portion of the neurons' outputs, thereby promoting the acquisition of resilient representations by the neural network. The second densely connected layer comprises 150 neurons, thereby augmenting the model's ability to capture intricate relationships within the dataset.

The output layer is comprised of five neurons, each representing one of the five classes involved in the

classification task. The activation function employed in this layer is the sigmoid activation, which is chosen for its ability to efficiently carry out multi-label classification by generating probability values for each individual class.

The model is trained using the categorical cross-entropy loss function, and the Adam optimizer is utilized to efficiently update the weights. The training procedure is supervised by employing early stopping techniques in order to avoid excessive training iterations and conserve computational resources. During the training process, the model acquires the ability to minimize the loss function and maximize accuracy, thereby enhancing its capacity to generalize effectively to data that has not been previously encountered. The performance of the model is assessed on a distinct validation dataset in order to evaluate its effectiveness on data that it has not been previously exposed to. The training procedure encompasses a maximum of 150 epochs, during which a batch size of 128 is employed in order to attain optimal accuracy. The utilization of a CNN architecture that incorporates larger filter sizes and denser layers is anticipated to enhance the model's ability to extract intricate features, thereby resulting in improved accuracy and resilience in the classification of images. The model under consideration demonstrates a high level of compatibility with a range of image recognition applications, encompassing medical imaging, object detection, and pattern recognition endeavors.

The CNN architecture is designed as follows:

1) Convolutional Layer 1:

$$O_1 = \text{Conv2D}(X, W_1) + b_1 \tag{47}$$
$$O_1 = \text{ReLU}(O_1) \tag{48}$$

2) Max-Pooling Layer 1:

$$O_2 = \text{MaxPool2D}(O_1) \tag{49}$$

3) Convolutional Layer 2:

$$O_3 = \text{Conv2D}(O_2, W_2) + b_2 \tag{50}$$
$$O_3 = \text{ReLU}(O_3) \tag{51}$$

4) Max-Pooling Layer 2:

$$O_4 = \text{MaxPool2D}(O_3) \tag{52}$$

5) Convolutional Layer 3:

$$O_5 = \text{Conv2D}(O_4, W_3) + b_3 \tag{53}$$
$$O_5 = \text{ReLU}(O_5) \tag{54}$$

6) Max-Pooling Layer 3:

$$O_6 = \text{MaxPool2D}(O_5) \tag{55}$$

7) Convolutional Layer 4:

$$O_7 = \text{Conv2D}(O_6, W_4) + b_4 \tag{56}$$
$$O_7 = \text{ReLU}(O_7) \tag{57}$$

8) Max-Pooling Layer 4:

$$O_8 = \text{MaxPool2D}(O_7) \tag{58}$$

9) Flattening Layer:

$$O_9 = \text{Flatten}(O_8) \tag{59}$$

10) Hidden Dense Layer 1 with Dropout:

$$O_{10} = \text{Dense}(O_9, W_5) + b_5 \tag{60}$$
$$O_{10} = \text{ReLU}(O_{10}) \tag{61}$$
$$O_{10} = \text{Dropout}(O_{10}) \tag{62}$$

11) Hidden Dense Layer 2:

$$O_{11} = \text{Dense}(O_{10}, W_6) + b_6 \tag{63}$$
$$O_{11} = \text{ReLU}(O_{11}) \tag{64}$$

12) Output Layer:

$$O_{12} = \text{Dense}(O_{11}, W_7) + b_7 \tag{65}$$
$$O_{12} = \text{Sigmoid}(O_{12}) \tag{66}$$

The model is trained using the training dataset (X_{train}, Y_{train}) and evaluated on the test dataset (X_{test}, Y_{test}) to assess its performance in image classification. Equations 47 through 66 give the architecture of the model used.

8) HYPERPARAMETER TUNING FOR CNN

The optimization of hyperparameters is a critical component in the development of efficient CNN structures for tasks pertaining to image classification. In the present context, a comprehensive tuning procedure was undertaken to enhance the performance of our CNN model. The hyperparameters specified in Table 5 are of utmost importance in the construction of an effective CNN model for image classification tasks. The practice of evaluating different hyperparameter values can yield advantageous results in identifying the most optimal configuration that yields the highest accuracy and generalization performance.

The tuning procedure encompasses the determination of optimal values for various hyperparameters, encompassing but not limited to the learning rate, batch size, number of filters, kernel size, pooling size, dropout rate, number of neurons, activation function, number of epochs, early stopping patience, and optimizer. The selected values were obtained through a thorough analysis, leading to the development of a CNN model that exhibits a significant level of accuracy and efficiency in the task of image classification. Table 5 presents the fundamental hyperparameters, accompanied by their corresponding explanations, potential ranges, and selected values.

C. EVALUATION METRICS

Various performance metrics, including recognition accuracy [54], F1-score [65], and Kappa (κ) score [66], were employed in the evaluation and analysis of the recognition system [67]. These metrics are commonly assessed using a confusion matrix. Here are some important terms associated with confusion metrics:

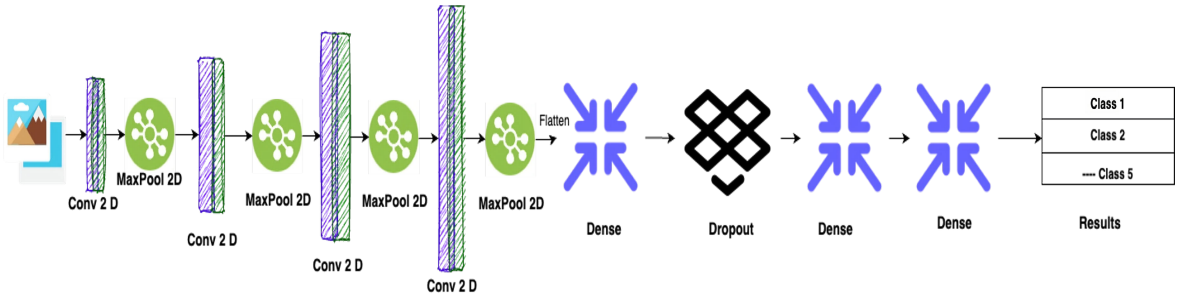


FIGURE 7. Architecture of CNN model implemented.

TABLE 5. Hyperparameter Tuning For Cnn Model.

Sl No	Hyperparameter	Description	Possible Values	Chosen Values
1	Learning Rate	Controls the step size during gradient descent.	[0.001, 0.01, 0.1]	0.001
2	Batch Size	Number of training examples used in one iteration.	[32, 64, 128]	128
3	Number of Filters	Number of filters in each convolutional layer.	[32, 64, 128, 256]	64, 128, 256, 512
4	Kernel Size	Size of the convolutional filter/kernel.	[(3, 3), (5, 5), (7, 7)]	(3, 3)
5	Pooling Size	Size of the max-pooling filter.	[(2, 2), (3, 3)]	(2, 2)
6	Dropout Rate	Probability of dropping a neuron during training.	[0.3, 0.5, 0.7]	0.5
7	Number of Neurons	Number of neurons in each hidden dense layer.	[100, 150, 200, 300]	400, 150
8	Activation Function	Activation function used in hidden layers.	ReLU, Tanh	ReLU
9	Number of Epochs	Number of times the entire dataset is passed through.	[50, 100, 150]	150
10	Early Stopping Patience	Number of epochs with no improvement to stop training.	[3, 5, 10]	3
11	Optimizer	Optimization algorithm used during training.	Adam, RMSprop, SGD	Adam

- A True Positive (TP) is a metric that denotes the accurate number of positive predictions [67].
- The True Negative (TN) metric accurately reflects the number of negative predictions that were correctly identified [67].
- A False Positive (FP) refers to a situation in which an incorrect number of positive predictions are specified [67].
- A False Negative (FN) occurs when the number of negative predictions is incorrect [67].

The present study delineates several performance metrics utilized in this research.

1) RECOGNITION ACCURACY

Recognition Accuracy (RA) can be defined as the proportion of correctly identified test samples in relation to the total number of input or test data samples. This can be expressed mathematically as:

$$RA = \frac{TP + TN}{TP + TN + FP + FN} \tag{67}$$

where TP, TN, FP, FN are as specified in section IV-C

2) PRECISION

Precision (P) is a performance measure that quantifies the ratio of accurately predicted positive samples to all samples predicted as positive. The mathematical representation is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{68}$$

where TP, FP are as specified in section IV-C.

3) F1 SCORE

The F1-score, also known as F1, is a widely used metric that integrates precision and recall to generate a unified measure. This methodology offers a comprehensive evaluation of the model's accuracy by assessing its capacity to correctly identify positive instances and encompass all pertinent positive instances. The F1-score is determined by calculating the harmonic mean of precision and recall, and it is formally denoted as:

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{69}$$

The concept of precision is addressed in section IV-C2, whereas recall, which is also referred to as sensitivity or true positive rate, quantifies the ratio of accurately predicted positive samples to the overall number of positive samples within the dataset. The statement pertains to the model's capacity to effectively capture all pertinent positive samples.

The mathematical representation of recall is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{70}$$

where TP, FP, FN are as specified in section IV-C.

4) AREA UNDER THE CURVE AND RECEIVER OPERATING CHARACTERISTIC

The assessment of the efficacy of a word image recognition model frequently depends on performance metrics, such as the Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC) values. The aforementioned metrics play a pivotal role in evaluating the performance of the

model. The ROC curve is a graphical representation that illustrates the trade-off between sensitivity (true positive rate) and specificity (1 - false positive rate) at various classification thresholds. On the other hand, the AUC is a metric that measures the overall performance of the recognition model by evaluating the area beneath the ROC curve. The ROC curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) on a graph. Every individual data point on the curve corresponds to a unique classification threshold.

The calculation of the AUC involves the integration of the ROC curve. The AUC is a quantitative measure utilized to evaluate the effectiveness of a classifier. The numerical value in question is confined to the interval between 0 and 1. A classifier with an AUC value of 0.5 indicates random performance, whereas a classifier with an AUC value of 1 denotes perfect performance.

AUC of the ROC curve is as follows:

$$\text{AUC} = \int_0^1 \text{TPR}(FPR)\, d(FPR) \tag{71}$$

where TPR(FPR) represents the True Positive Rate (sensitivity) as a function of the False Positive Rate (FPR) at different classification thresholds. The integral is taken over the entire range of FPR, which is from 0 to 1. The AUC provides a single scalar value that quantifies the overall performance of the classification model, with higher values indicating better performance.

5) ADDITIONAL METRICS

Supplementary characteristics such as Model Build Time and Model Run Time were also considered. Model Build Time (MBT) is the duration required to build or train a machine learning model, measured as the difference between the end and start times of the construction process. Model Run Time is the duration necessary for a trained model to generate predictions on new data, calculated as the difference between the end and beginning times of the prediction process. Each metric is measured in temporal units including seconds, minutes, and hours.

- Model Build Time:

$$\text{MBT} = \text{End Build Time} - \text{Start Build Time} \tag{72}$$

where: -End Build Time represents the time when the model construction or training process is completed.
-Start Build Time represents the time when the model construction or training process started.

- Model Run Time:

$$\text{MRT} = \text{End Prediction Time} - \text{Start Prediction Time} \tag{73}$$

where:
- End Run Time represents the time when the model finishes processing the input data and generates the predictions.

TABLE 6. Classification results (LR) for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Logistic Regression	0.956	0.948	0.956	0.926
Accuracy	0.956	0.948	0.957	0.926
Kappa	0.946	0.934	0.946	0.908
F1-score	0.956	0.948	0.956	0.926
Build Time (seconds)	3.195	2.362	6.770	3.889
Run Time (seconds)	0.004	0.001	0.003	0.002

TABLE 7. Classification results DT for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Decision Tree	0.748	0.723	0.782	0.721
Accuracy	0.748	0.723	0.782	0.721
Kappa	0.685	0.650	0.728	0.652
F1-score	0.749	0.723	0.782	0.721
Build Time (seconds)	1.250	0.416	1.176	1.101
Run Time (seconds)	0.002	0.001	0.003	0.003

TABLE 8. Classification results (random forest) for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Random Forest	0.925	0.927	0.940	0.918
Accuracy	0.925	0.927	0.940	0.918
Kappa	0.906	0.907	0.925	0.898
F1-score	0.925	0.927	0.940	0.918
Build Time (seconds)	2.473	0.755	2.558	1.945
Run Time (seconds)	0.027	0.011	0.035	0.024

TABLE 9. Classification results (svm linear kernel) for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Accuracy	0.926	0.927	0.940	0.905
Precision	0.926	0.927	0.941	0.905
Kappa	0.907	0.908	0.926	0.881
F1-score	0.926	0.927	0.940	0.904
Build Time (seconds)	15.058	9.162	54.358	14.752
Run Time (seconds)	0.005	0.001	0.006	0.003

- Start Run Time represents the time when the model starts processing the input data.

V. EXPERIMENTAL RESULTS

According to Section III-B, the investigation performed experiments on four datasets derived from five literary works. Uniform classification algorithms were employed, utilizing the implemented features described in Section IV-A. Preliminary results were presented, facilitating dataset comparison to identify the most suitable one for future experimentation. The classifiers used for performance analysis are described in Section IV-B. Based on the evaluation criteria outlined in Section IV-C, a comprehensive understanding of the optimal combination of techniques for automated word recognition was obtained. The experimentation employed a 70%-30% split for training and testing tasks.

The results of the classification performed using LR, as presented in Table 6, demonstrate that Dataset 3 exhibited superior performance. This is evident from its accuracy, precision, F1-score, and Kappa coefficient, all of which were recorded at a value of 0.956. The building time for Dataset 3 is 6.770 seconds, while the execution time is 0.003 seconds.

TABLE 10. Classification results (svm rbf) for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Accuracy	0.954	0.964	0.956	0.927
Precision	0.956	0.965	0.959	0.935
Kappa	0.942	0.955	0.945	0.908
F1-score	0.954	0.964	0.956	0.928
Build Time (seconds)	10.979	1.730	13.804	8.525
Run Time (seconds)	4.264	0.715	6.161	3.253

TABLE 11. Classification results (KNN) for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Accuracy	0.712	0.634	0.712	0.631
Precision	0.774	0.684	0.812	0.702
Kappa	0.640	0.524	0.640	0.539
F1-score	0.704	0.607	0.712	0.592
Build Time (seconds)	0.002	0.001	0.003	0.002
Run Time (seconds)	0.324	0.062	0.582	0.251

TABLE 12. Classification results (GB) for different datasets.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Accuracy	0.954	0.953	0.960	0.943
Precision	0.956	0.965	0.959	0.935
Kappa	0.943	0.941	0.950	0.928
F1-score	0.954	0.953	0.960	0.942
Build Time (seconds)	165.144	62.380	159.335	143.198
Run Time (seconds)	0.027	0.010	0.034	0.021

TABLE 13. Classification results for different datasets using CNN model.

Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Accuracy	0.94	0.854	0.972	0.933
Precision	0.942	0.878	0.972	0.936
Kappa	0.925	0.816	0.965	0.916
F1-score	0.94	0.851	0.972	0.933
Build Time (seconds)	84.294	39.455	122.359	95.737
Run Time (seconds)	0.558	0.332	0.74	0.529

FIGURE 8. Results of LR classifier for all Datasets.

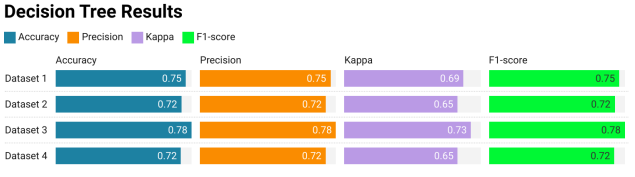


FIGURE 9. Results of DT classifier for all datasets.

Figure 8 and 16 show the classification and model times for the same.

According to the results obtained from the DT algorithm (refer to Table 7), it can be observed that Dataset 3 exhibits an F1-score, accuracy, and precision of 0.782, along with a

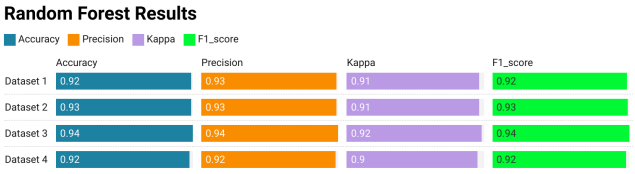


FIGURE 10. Results of RF classifier for all datasets.

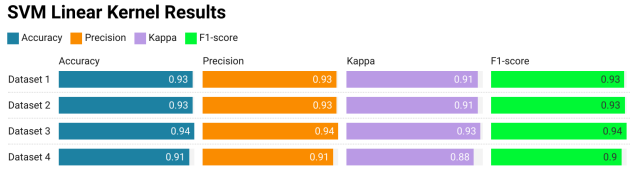


FIGURE 11. Results of SVM linear kernel classifier for all Datasets.

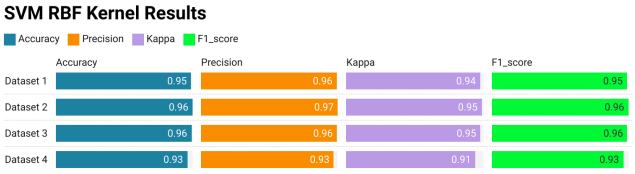


FIGURE 12. Results of SVM RBF kernel classifier for all datasets.

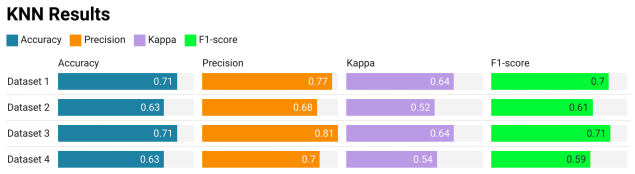


FIGURE 13. Results of KNN classifier for all datasets.

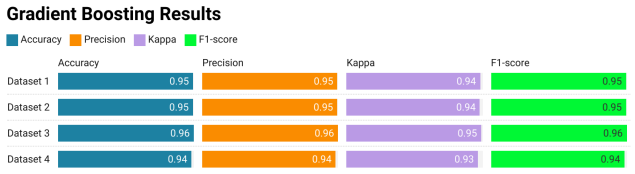


FIGURE 14. Results of GB classifier for all datasets.

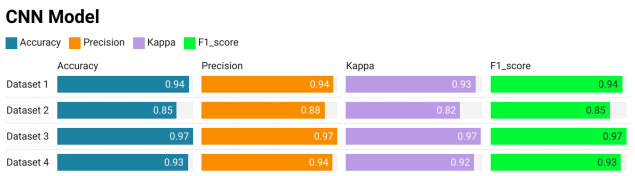


FIGURE 15. Results of CNN for all datasets.

Kappa coefficient of 0.728. The construction time for Dataset 3 is 1.176 seconds, while the execution time is 0.003 seconds. Figure 9 and 17 depict the classification and model times for the same.

TABLE 14. Results for dataset 3.

Metric	LR	DT	RF	SVM Linear	SVM RBF	KNN	GB	CNN
Accuracy	**0.957**	0.782	0.940	0.941	**0.957**	0.712	**0.961**	0.973
Precision	**0.957**	0.782	0.940	0.941	**0.959**	0.941	**0.961**	0.973
Kappa	0.943	0.728	0.925	0.926	**0.945**	0.640	**0.951**	0.965
F1_score	**0.957**	0.782	0.940	0.941	**0.957**	0.712	**0.961**	0.973
AUC Scores	0.973	0.857	0.964	0.972	0.973	0.807	0.975	0.996
Build Time (seconds)	6.771	1.177	2.558	54.358	13.804	0.003	159.335	122.359
Run Time (seconds)	0.003	0.004	0.036	0.006	6.161	0.582	0.035	0.74

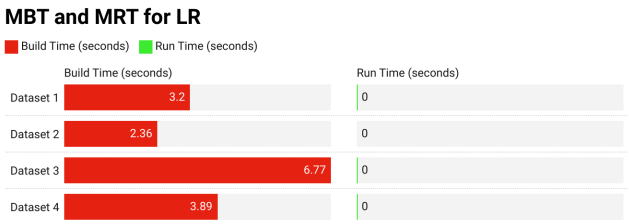


FIGURE 16. Build and run times for logistic regression.

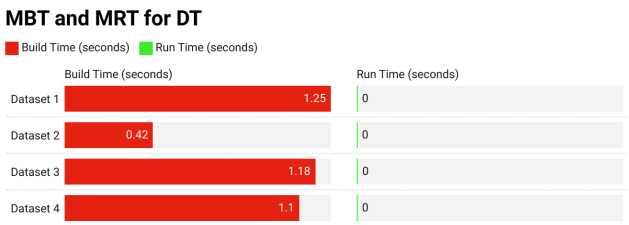


FIGURE 17. Build and run times for decision tree.

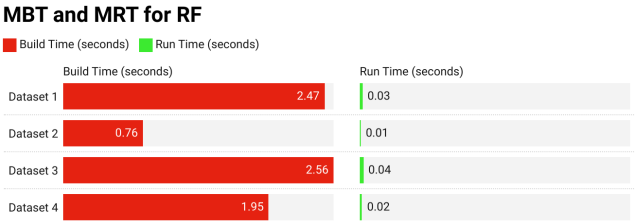


FIGURE 18. Build and run times for random forest.

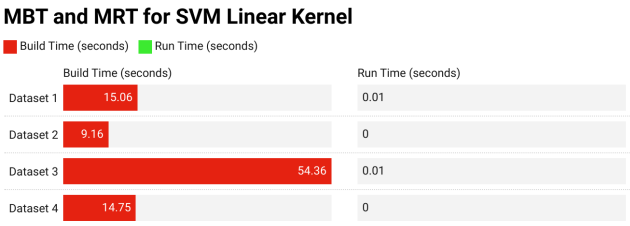


FIGURE 19. Build and run times for svm linear kernel.

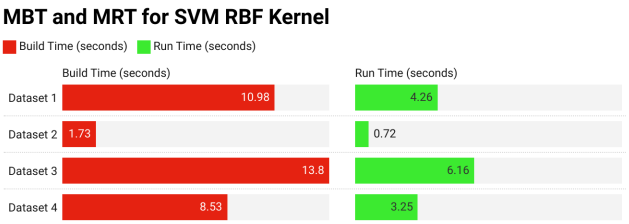


FIGURE 20. Build and run times for SVM RBF kernel.

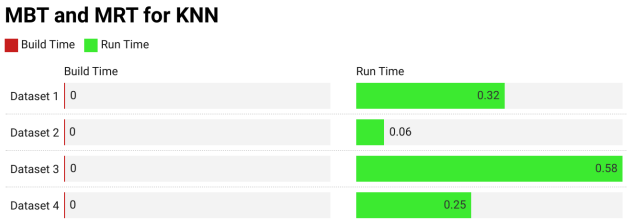


FIGURE 21. Build and run times for KNN.

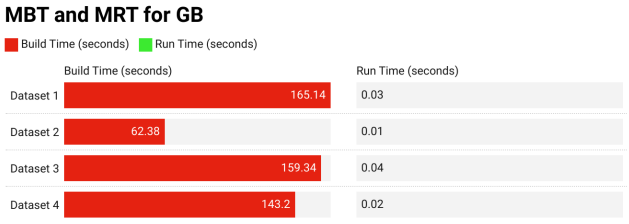


FIGURE 22. Build and run times for GB.

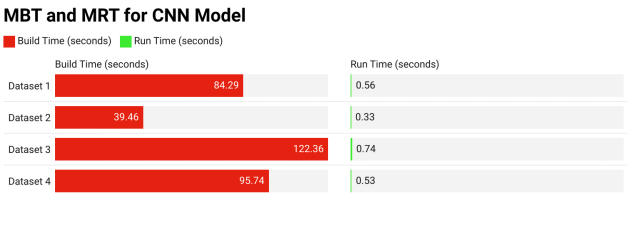


FIGURE 23. Build and run times for CNN.

According to the RF algorithm, as shown in Table 8, Dataset 3 demonstrated a precision, accuracy, and F1-score of 0.940. The construction time for Dataset 3 is 2.558 seconds, whereas the execution time is 0.035 seconds. Figure 10 and 18 depict the classification and model execution times for the same.

Based on the performance metrics of SVM Linear Kernel presented in Table 9, it can be observed that Dataset 3 exhibits the highest accuracy score of 0.940, whereas

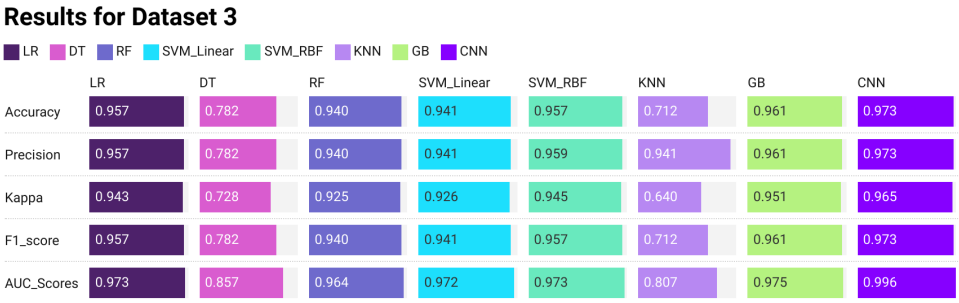


FIGURE 24. Classification results for dataset 3.

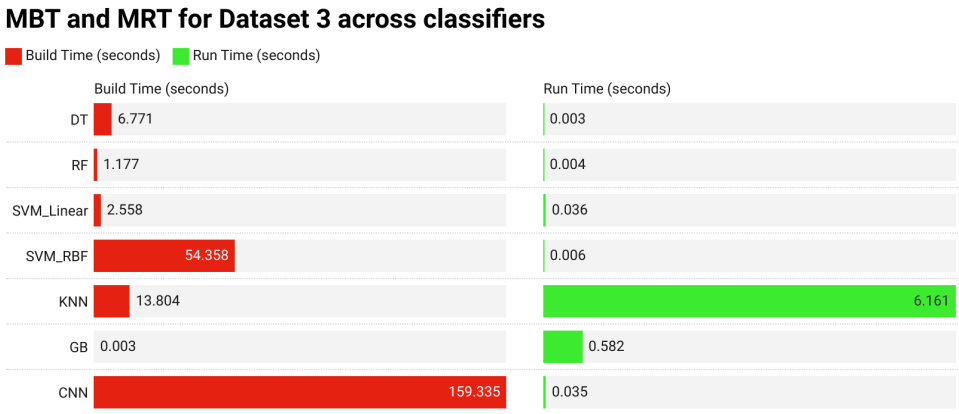


FIGURE 25. Build and run times for all classifiers for dataset 3.

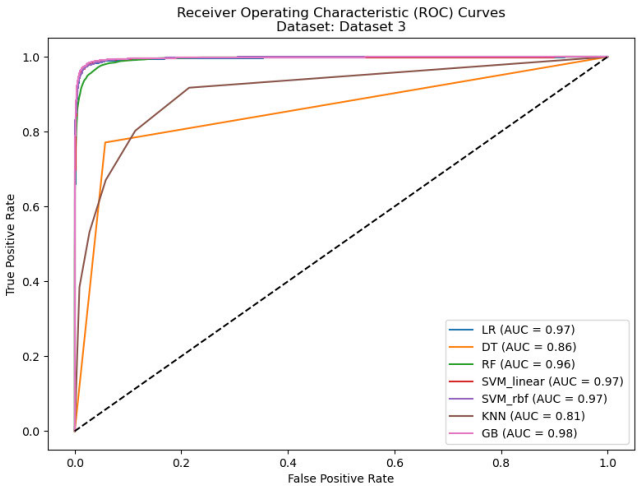


FIGURE 26. ROC values across classifiers.

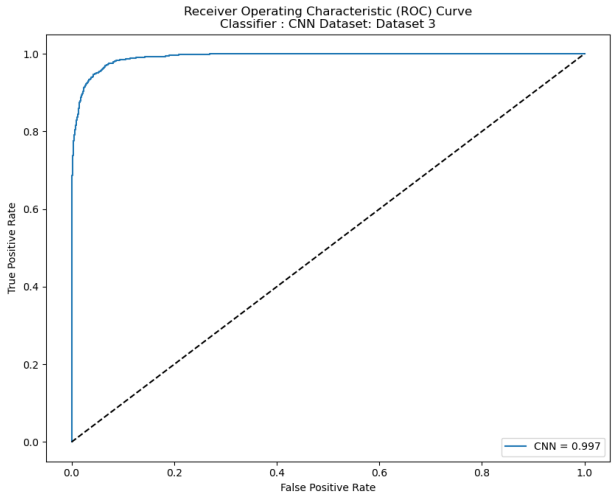


FIGURE 27. ROC values for CNN.

Dataset 4 demonstrates the lowest accuracy score of 0.905. Dataset 2 exhibits the minimum time taken to build of 9.162 seconds, whereas Dataset 3 demonstrates the maximum build time of 54.358 seconds. Figure 11 and 19 represent the classification and model times for the same.

The results of the classification using SVM with RBF Kernel, as presented in Table 10, indicate that Dataset 2 exhibits the highest accuracy rate of 0.964, while Dataset 4 demonstrates the lowest accuracy rate of 0.927. The construction and execution duration for Dataset 2 are 1.73 and 0.71 seconds, correspondingly (refer Fig 12 and 20).

The KNN algorithm, as shown in Table 11, indicates that Dataset 3 exhibits superior performance compared to other

TABLE 15. Comparison of methods (feature based).

S. No.	Authors	Scripts	Word Count	Feature Set	Classifier	RA (%)
1	Shaw et al. [68]	Devanagari	39,700	Directional chain code	HMM	80.2
2	Shaw et al. [69]	Devanagari	39,700	Stroke based	HMM	84.31
3	Shaw et al. [70]	Devanagari	39,700	Combination of skeleton and contour based	SVM	79.01
4	Shaw et al. [71]	Devanagari	39,700	DDD and GSC	Multi-class SVM	88.75
5	Shaw and Parui [72]	Devanagari	13,000	Stroke based (Stage-1); Wavelet (Stage-2)	HMM (Stage-1);	91.25
6	Singh et al. [73]	Devanagari	28,500	Curvelet transform	SVM and KNN	85.6 (SVM); 93.21 (KNN)
7	Kumar [74]	Devanagari	>3500	Neighbor pixel weight and gradient feature	MLP	80.8
8	Parui and Shaw [75]	Devanagari	10,000	Stroke based	HMM	87.71
9	Singh [54]	Devanagari	20,000	Combination of uniform zoning, diagonal and centroid features	Gradient boosted decision tree	94.33
10	Ramachandrula [76]	Hindi	39,600	Directional element	Dynamic Programming	91.23
11	Malakar et al. [77]	Hindi	4,620	Low-level features	MLP	96.82
12	Kaur and Kumar [78]	Gurumukhi	40,000	Zoning features	XGBoost	91.66
13	Ghosh et al. [79]	Bangla	7,500	Gradient features and modified SCF; MA-based wrapper filter selection approach	MLP	93
14	Malakar et al. [80], [81]	Bangla	12,000	Gradient-based and elliptical	MLP	95.3
15	Bhunia et al. [82]	Bangla, Devanagari, Gurumukhi	3,856; 3,589; 3,142	PHOG feature	HMM (Middle-zone), SVM (Upper/Lower zone)	>60
16	Proposed method (Feature-Based)	Assamese	8,561	Combination of Multiple Low level - Shape, Region, Descriptor based features	Gradient Boosting, SVM (RBF Kernel)	96.1 (GB) & 95.7 (SVM)

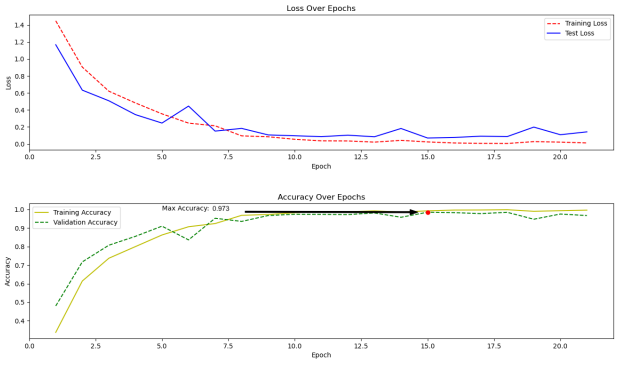


FIGURE 28. Loss vs epochs and accuracy vs epochs.

datasets in terms of precision, accuracy, kappa, and F1-score. The results can be visualised through Figure 13 and 22.

The GB Classifier was evaluated in datasets 1 to 4. Dataset 1 exhibited a kappa coefficient of 0.943, along with an accuracy, precision, and F1-score of 0.954. The construction process required a total of 165.144 seconds, while the execution phase took a mere 0.027 seconds. Dataset 2 exhibited notable levels of accuracy, precision, F1-score, and kappa coefficient. The construction process lasted for a duration of 62.380 seconds, while the operational phase required a mere 0.010 seconds. Dataset 3 exhibited a kappa coefficient of 0.950, along with accuracy, precision, and F1-score values of 0.960. The build process required a total of 159.335 seconds, while the execution of the program took 0.034 seconds. Dataset 4 exhibited diminished levels of accuracy, precision, F1-score, and kappa coefficient, registering values of 0.943 and 0.928, respectively. The construction process lasted for a duration of 143.198 seconds, while the execution phase had a duration of 0.021 seconds.

The study employed various machine learning algorithms, including LR,DT,RF,SVM,KNN,GB to conduct experiments on four distinct datasets. Among the algorithms considered, it was observed that Dataset 3, comprising a total of 8561 images and 1523 features extracted from 19 distinct categories, demonstrated exceptional performance across multiple evaluation metrics. The study exhibited significant effectiveness and displayed potential for attaining optimal classification results.

Nevertheless, it is crucial to acknowledge that gradient boosting, despite its promising outcomes (refer to Fig 14)), exhibited a considerable duration for constructing the model (refer to Fig 22), which may pose a limitation in specific situations. However, the gradient boosting classifier exhibited notable levels of accuracy, precision, F1-score, and computational efficiency when applied to various datasets.

A. PERFORMANCE ANALYSIS FOR DATASET 3

However, it is important to recognize that gradient boosting, despite its favorable results (see Fig 14), demonstrated a significant time requirement for model construction (see Figure 22), which could present a constraint in certain scenarios. Nevertheless, the gradient boosting classifier demonstrated significant levels of accuracy, precision, F1-score, and computational efficiency when employed on diverse

(a) Confusion Matrix for GB

(b) Confusion Matrix for LR

(c) Confusion Matrix for SVM (RBF)

(d) Confusion Matrix for CNN

FIGURE 29. **Confusion matrices.**

datasets. Based on the obtained outcomes, it can be inferred that the Support Vector Machine algorithm utilizing the Radial Basis Function Kernel exhibits promise as a viable option for classification tasks in close proximity to LR (see Figure 8 and Figure 16). Additional information and outcomes pertaining to Support Vector Machines utilizing the Radial Basis Function Kernel can be accessed in Figure 12 and Figure 20. It is noteworthy to mention that Dataset 3 consistently exhibited superior performance compared to other datasets. To enhance comprehension, the findings of Dataset 3 have been isolated for the purpose of facilitating understanding in Table 14.

The confusion matrices, as illustrated in the figure 28, offer valuable insights into the performance of the models. The analysis demonstrates that GB exhibits superior performance compared to LR and SVM-RBF in the context of machine learning algorithms. Additionally, CNN demonstrates the highest levels of accuracy and precision, positioning it as a highly promising option for a wide range of classification tasks. The aforementioned findings demonstrate the potential efficacy of employing advanced machine learning techniques for addressing practical challenges in various domains.

This study conducts an investigation into different classification algorithms and determined that Dataset 3 demonstrated superior performance. Although gradient boosting demonstrated promising outcomes, it is important to acknowledge that the time required for model building could potentially serve as a constraint. Hence, it is advisable to employ Support Vector Machines with Radial Basis Function Kernel as a viable alternative for classification purposes if using Classical ML algorithms. However, CNN leads in overall accuracy values. Figure 27 presents a comprehensive examination of the relationship between loss and epoch, as well as accuracy and epoch, within the context of training a CNN model. The figures provided depict the progressive convergence of the model during the training phase, as indicated by the declining loss and improving accuracy observed with each epoch. The importance of choosing the correct epoch is underscored in this extensive examination, as it plays a crucial role in achieving the best possible performance of the model.

TABLE 16. Comparison of methods (CNN based).

S. No.	Authors	Scripts	Word Count	Feature Set	Classifier	RA (%)
1	Nurseitov et al. [83]	Russian, Kazakh	1400 filled forms	CNN for feature extraction, MLP for word classification	CNN, MLP	55.3 (CNN); 57.1 (SimpleHTR); 58.3 (Beamsearch); 75.1 (Wordbeam-search)
2	Bulla et al. [84]	Handwritten words (JPEG Compressed Domain)	39,700	CNN and BiLSTM	CNN, BiLSTM	89.05
3	Sudholt and Fink [64]	English	–	Attribute CNNs	CNN	State-of-the-art results
4	Zhang et al. [85]	Chinese Characters	–	RNN (LSTM and GRU)	RNN (LSTM and GRU)	State-of-the-art results
5	Proposed method (CNN)	Assamese	8,561	CNN based	CNN with multiple layers	97.3

B. PERFORMANCE ANALYSIS WITH OTHER STUDIES

The present study conducts a comparative analysis of various methods employed in different script contexts with the aim of enhancing knowledge sharing and dissemination. Please refer to Table 15 for additional information. Various methodologies and classifiers have demonstrated word recognition accuracies spanning from 80.2% to 96.8% in scripts such as Devanagari, Hindi, Bangla, and Gurumukhi. With the utilization of low-level features and the application of either GB or SVM (RBF Kernel) classifiers, the Assamese script feature set demonstrated a notable accuracy of 96.1%. Table 15 gives an account of the performance of the curated featureset with other existing works in different scripts like Devanagari, Hindi, Bangla etc. The table 16 provides a comparative analysis of various approaches utilizing CNN for tasks related to text processing. The initial approach, as suggested by Nurseitov et al., centers on the examination of Russian and Kazakh scripts through the utilization of 1400 completed forms [83].The proposed approach utilizes CNN for feature extraction and Multilayer Perceptron (MLP) for word classification. The experimental results demonstrate recognition accuracies of 55.3% (CNN), 57.1% (SimpleHTR), 58.3% (Beamsearch), and 75.1% (Wordbeamsearch). The second approach, as presented by Bulla et al., focuses on handwritten words in the JPEG Compressed Domain, encompassing a total word count of 39,700 [84]. The feature set employed in this study consists of CNN and Bidirectional Long Short-Term Memory (BiLSTM). The experimental results demonstrate a recognition accuracy of 89.05%. The experimental findings indicate that the recognition accuracy is 89.05%. In their study, Sudholt and Fink [64] utilized Attribute CNNs as the feature set and employed a CNN classifier to analyze English scripts. The approach employed by the researchers resulted in noteworthy outcomes, establishing it as the predominant method within the field. The research conducted by Zhang et al. focuses on the examination of Chinese characters using recurrent neural networks, specifically long short-term memory and gated recurrent unit, as the feature set [85]. The researchers have achieved significant and innovative results in their investigation. The methodology proposed in this research has been effectively implemented for the analysis of Assamese scripts, comprising a comprehensive corpus of 8,561 words. The method utilized in this research comprises several layers of CNN, leading to a significant recognition accuracy of 97.3%.

VI. DISCUSSIONS

A range of machine learning algorithms are employed to analyze and classify 19 characteristics that are related to shape. Various performance metrics, including Accuracy, Precision, Kappa, F1-score, and computational efficiency, are utilized to assess the effectiveness of these methods. Among the algorithms evaluated on Item Dataset 3, GB demonstrates a notable accuracy of 96.03%, with LR and SVM with RBF kernel closely trailing behind. Although Gradient Boosting achieves a notable accuracy rate of 96.03%, it is crucial to recognize the limitation it possesses in terms of extended model construction duration. Additionally, the study demonstrates the effectiveness of Convolutional Neural Network (CNN) in identifying words based on their shape, with a high accuracy rate of 97.3% and an excellent AUC score of 0.996. CNN's outstanding performance confirms its potential as a potent tool for language preservation and document digitization, demonstrating promising results for future advancements in word image processing in Indic languages. Nevertheless, it is important to acknowledge that the construction of the model may not be required on every occasion, unless there is a specific requirement to retrain it using fresh data. The aforementioned discoveries make a significant contribution to the progress of word recognition technology in Indic languages, with a specific focus on the preservation and enhancement of the Assamese language and its literary works.

VII. CONCLUSION

This study examines the use of shape-based methods and machine learning algorithms to identify Assamese words in Indic languages. The objective is to protect language and convert documents into digital format, thus ensuring the preservation of linguistic and cultural heritage. Dataset 3, consisting of 8,561 images, demonstrated superior

performance in four dataset experiments. The accuracy rates of the models were as follows: GB achieved the highest accuracy rate of 96.03%, followed by LR and SVM with RBF kernel at 95.64% and 95.60%, respectively. These results were obtained using a total of 1523 features. The CNN model achieved outstanding results, with an accuracy of 97.3% and an impressive AUC score of 0.996. The algorithms successfully identify Assamese words in shape-based word identification, showing potential for language preservation and document digitization.

VIII. FUTURE SCOPE

This section provides an analysis of the potential future expansion of shape-based word recognition techniques in the context of Assamese and other Indic languages. This statement proposes potential avenues for further investigation.

Determining the best shape-oriented characteristics: Future research can identify the best shape-based features for word recognition in Assamese and other Indic languages. Recognization systems can be improved by examining unique traits.

Determining the ensemble techniques: Further research is warranted to investigate the potential of utilizing ensemble techniques to enhance shape-based word recognition in Assamese and other Indic languages. Ensemble methods, such as Catboost and Stacking, have demonstrated their efficacy in integrating multiple shape-based recognition models, leading to improved overall accuracy and robustness. Ensemble methods have the potential to enhance performance and generalization on complex datasets by harnessing the diversity demonstrated by individual models.

REFERENCES

[1] S. Nigam, S. Verma, and P. Nagabhushan, "Document analysis and recognition: A survey," *J. Latex Class Files*, vol. 14, no. 8, Aug. 2021.

[2] M. Liwicki and H. Bunke, "IAM-OnDB–an on-line English sentence database acquired from handwritten text on a whiteboard," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 956–961.

[3] U.-V. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, Jun. 1999, pp. 705–708.

[4] U.-V. Marti and H. Bunke, "The IAM-database: An English sentence database for offline handwriting recognition," *Int. J. Document Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, Nov. 2002.

[5] Y. LeCun, C. Cortes, and C. Burges. *The MNIST Database of Handwritten Digits*. Accessed: Jun. 5, 2023. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[6] A. Mezghani, S. Kanoun, M. Khemakhem, and H. E. Abed, "A database for Arabic handwritten text image recognition and writer identification," in *Proc. Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2012, pp. 399–402.

[7] N. B. Amara, O. Mazhoud, N. Bouzrara, and N. Ellouze, "ARABASE: A relational database for Arabic OCR systems," *Int. Arab J. Inf. Technol.*, vol. 2, no. 4, pp. 259–266, 2005.

[8] R. Safabaksh, A. R. Ghanbarian, and G. Ghiasi, "HaFT: A handwritten Farsi text database," in *Proc. 8th Iranian Conf. Mach. Vis. Image Process. (MVIP)*, Sep. 2013, pp. 89–94.

[9] A. M. Bidgoli and M. Sarhadi, "IAUT/PHCN: Islamic Azad University of Tehran/Persian handwritten city names, a very large database of handwritten Persian word," in *Proc. ICFHR*, vol. 11, 2008, pp. 192–197.

[10] T. Subramaniam, U. Pal, H. Premaretne, and N. Kodikara, "Holistic recognition of handwritten Tamil words," in *Proc. 3rd Int. Conf. Emerg. Appl. Inf. Technol.*, Nov. 2012, pp. 165–169.

[11] T. Mondal, U. Bhattacharya, S. K. Parui, K. Das, and D. Mandalapu, "On-line handwriting recognition of Indian scripts—The first benchmark," in *Proc. 12th Int. Conf. Frontiers Handwriting Recognit.*, Nov. 2010, pp. 200–205.

[12] E. Grosicki, M. Carré, J.-M. Brodin, and E. Geoffrois, "Results of the RIMES evaluation campaign for handwritten mail processing," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 231–235.

[13] E. Kavallieratou, N. Liolios, E. Koutsogeorgos, N. Fakotakis, and G. Kokkinakis, "The GRUHD database of Greek unconstrained handwriting," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 561–565.

[14] M. Nakagawa and K. Matsumoto, "Collection of on-line handwritten Japanese character pattern databases and their analyses," *Document Anal. Recognit.*, vol. 7, no. 1, pp. 1–10, Mar. 2004, doi: 10.1007/s10032-004-0125-4.

[15] D. H. Kim, Y. S. Hwang, S. T. Park, E. J. Kim, S. H. Paek, and S. Y. Bang, "Handwritten Korean character image database PE92," in *Proc. 2nd Int. Conf. Document Anal. Recognit.*, 1993, pp. 470–473.

[16] R. A. Wilkinson, "The first census optical character recognition system conference," U.S. Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 184, 1992.

[17] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994, doi: 10.1109/34.291440.

[18] M. W. A. Kesiman, J.-C. Burie, G. N. M. A. Wibawantara, I. M. G. Sunarya, and J.-M. Ogier, "AMADI_LontarSet: The first handwritten balinese palm leaf manuscripts dataset," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 168–173.

[19] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-DataBase: An off-line database for writer retrieval, writer identification and word spotting," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 560–564.

[20] S. Impedovo, G. Facchini, and F. M. Mangini, "A new cursive basic word database for bank-check processing systems," in *Proc. 10th IAPR Int. Workshop Document Anal. Syst.*, Mar. 2012, pp. 450–454.

[21] U. Bhattacharya, S. Parui, B. Shaw, and K. Bhattacharya, "Neural combination of ANN and HMM for handwritten Devanagari numeral recognition," in *Proc. 10th Int. Workshop Frontiers Handwriting Recognit.*, 2006, pp. 39–44.

[22] S. Al-Ma'adeed, D. Elliman, and C. A. Higgins, "A data base for Arabic handwritten text recognition research," *Int. Arab J. Inf. Technol.*, vol. 1, no. 1, pp. 1–5, Jan. 2004.

[23] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IFN/ENIT-database of handwritten Arabic words," in *Proc. CIFED*, vol. 2, 2002, pp. 127–136.

[24] L. Schomaker and M. Bulacu, "Automatic writer identification using connected-component contours and edge-based features of uppercase western script," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 787–798, 2004.

[25] K. Medhi, "Assamese grammar and origin of the Assamese language," Publication Board, Guwahati, India, Tech. Rep., 1988.

[26] J. J. Deka and A. T. Boro, "Charyapads as the oldest written specimen of Assamese literature," *Int. J. health Sci.*, vol. 10, pp. 7028–7034, May 2022.

[27] Encyclopædia Britannica. (May 8, 2023). *Assamese Language. Encyclopædia Britannica*. [Online]. Available: https://www.britannica.com/topic/Assamese-languag

[28] I. Hussain, N. Saharia, and U. Sharma, "Development of Assamese WordNet," in *Machine Intelligence: Recent Advances*. Chennai, India: Narosa Publishing House, 2011., pp. 272–280.

[29] B. Sarma and B. S. Purkayastha, "An affix based word classification method of Assamese text," *Int. J. Adv. Res. Comput. Sci.*, vol. 4, no. 9, pp. 1–12, 2013.

[30] O. Borgohain, P. Kumar, and S. Sutradhar, "Recognition of handwritten Assamese characters," in *Proc. 3rd Int. Conf. Artif. Intell.* Singapore: Springer, 2023, pp. 223–230.

[31] A. Choudhury and K. K. Sarma, "Trajectory-based recognition of in-air handwritten Assamese words using a hybrid classifier network," *Int. J. Document Anal. Recognit. (IJDAR)*, pp. 1–26, Jan. 2023.

[32] R. Das and T. D. Singh, "Assamese news image caption generation using attention mechanism," *Multimedia Tools Appl.*, vol. 81, no. 7, pp. 10051–10069, Mar. 2022.

[33] A. Choudhury and K. K. Sarma, "A CNN-LSTM based ensemble framework for in-air handwritten Assamese character recognition," *Multimedia Tools Appl.*, vol. 2021, pp. 1–36, Nov. 2021.

[34] K. Medhi* and S. K. Kalita, "Assamese handwritten character recognition using supervised fuzzy logic," *Int. J. Recent Technol. Eng.*, vol. 8, no. 5, pp. 3750–3758, Jan. 2020.

[35] S. Mandal, H. Choudhury, S. R. M. Prasanna, and S. Sundaram, "DNN-HMM based large vocabulary online handwritten Assamese word recognition system," in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Aug. 2018, pp. 321–326.

[36] U. Baruah and S. M. Hazarika, "A dataset of online handwritten Assamese characters," *J. Inf. Process. Syst.*, vol. 11, no. 3, pp. 325–341, 2015.

[37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.

[38] S. P. Deore, "HDWR_SmartNet: A smart handwritten Devanagari word recognition system using deep ResNet-based on scan profile method," in *Data Science*. Boca Raton, FL, USA: CRC Press, 2022, pp. 57–77.

[39] J. Žunić, K. Hirota, and P. L. Rosin, "A Hu moment invariant as a shape circularity measure," *Pattern Recognit.*, vol. 43, no. 1, pp. 47–57, Jan. 2010.

[40] O. E. Ghachi, A. Lakehal, and N. Labani, "Arabic hand print character recognition using new hybrid descriptor," *Int. J. Apllied Math.*, vol. 32, no. 1, pp. 1–6, Mar. 2019.

[41] H. Ren and Z.-N. Li, "Object detection using edge histogram of oriented gradient," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4057–4061.

[42] J. J. Koenderink and A. J. Van Doorn, "Surface shape and curvature scales," *Image Vis. Comput.*, vol. 10, pp. 557–564, Feb. 1992, doi: 10.1016/0262-8856(92)90076-F.

[43] Y.-Y. Liu, M. Chen, H. Ishikawa, G. Wollstein, J. S. Schuman, and J. M. Rehg, "Automated macular pathology diagnosis in retinal OCT images using multi-scale spatial pyramid and local binary patterns in texture and shape encoding," *Med. Image Anal.*, vol. 15, no. 5, pp. 748–759, Oct. 2011.

[44] H. Freeman, "On the encoding of arbitrary geometric configurations," *IEEE Trans. Electron. Comput.*, vol. EC-10, no. 2, pp. 260–268, Jun. 1961.

[45] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geographic Inf. Geovisualization*, vol. 10, no. 2, pp. 112–122, Dec. 1973.

[46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "33.3: Finding the convex hull," in *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 1990, pp. 955–956.

[47] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Inf. Process. Lett.*, vol. 1, pp. 132–133, Jan. 1972.

[48] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, Dec. 1996.

[49] L. Dong, J. Zhou, and Y. Y. Tang, "Noise level estimation for natural images based on scale-invariant kurtosis and piecewise stationarity," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 1017–1030, Feb. 2017.

[50] Y. Bai, L. Guo, L. Jin, and Q. Huang, "A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3305–3308.

[51] M. Saberioon, P. Císař, L. Labbé, P. Souček, P. Pelissier, and T. Kerneis, "Comparative performance analysis of support vector machine, random forest, logistic regression and k-nearest neighbours in rainbow trout (*Oncorhynchus Mykiss*) classification using image-based features," *Sensors*, vol. 18, no. 4, p. 1027, Mar. 2018.

[52] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008.

[53] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *J. Chemometrics*, vol. 18, no. 6, pp. 275–285, Jun. 2004.

[54] S. Singh, N. K. Garg, and M. Kumar, "On the performance analysis of various features and classifiers for handwritten devanagari word recognition," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7509–7527, Apr. 2023.

[55] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[56] V. R. Madireddy, "Content based image classification using support vector machine algorithm," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 1, pp. 1–5, Jan. 2018.

[57] S. Roopashree, J. Anitha, T. R. Mahesh, V. V. Kumar, W. Viriyasitavat, and A. Kaur, "An IoT based authentication system for therapeutic herbs measured by local descriptors using machine learning approach," *Measurement*, vol. 200, Aug. 2022, Art. no. 111484.

[58] E. Fix, *Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties*, vol. 1. Air Force Academy, CO, USA: USAF School of Aviation Medicine, 1985.

[59] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[60] S. C. Yenigalla, S. Rao, and P. S. Ngangbam, "Implementation of content-based image retrieval using artificial neural networks," *Eng. Proc.*, vol. 34, no. 1, p. 25, 2023.

[61] S. Sikandar, R. Mahum, and A. Alsalman, "A novel hybrid approach for a content-based image retrieval using feature fusion," *Appl. Sci.*, vol. 13, no. 7, p. 4581, Apr. 2023.

[62] I. G. P. S. Wijaya, A. Y. Husodo, and I. W. A. Arimbawa, "Real time face recognition based on face descriptor and its application," *TELKOMNIKA, Telecommun. Comput. Electron. Control*, vol. 16, no. 2, pp. 739–746, 2018.

[63] M. Okawa, "Online signature verification using single-template matching with time-series averaging and gradient boosting," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107227.

[64] S. Sudholt and G. A. Fink, "Attribute CNNs for word spotting in handwritten documents," *Int. J. Document Anal. Recognit.*, vol. 21, no. 3, pp. 199–218, Sep. 2018.

[65] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F_1 and macro-averaged F_1 scores," *Int. J. Speech Technol.*, vol. 52, no. 5, pp. 4961–4972, Mar. 2022.

[66] Y. Um, "Assessing classification accuracy using Cohen's Kappa in data mining," *J. Korea Soc. Comput. Inf.*, vol. 18, no. 1, pp. 177–183, Jan. 2013.

[67] A. K. Jain, P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000, doi: 10.1109/34.824819.

[68] B. Shaw, S. K. Parui, and M. Shridhar, "A segmentation based approach to offline handwritten Devanagari word recognition," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2008, pp. 256–257.

[69] B. Shaw, S. K. Parui, and M. Shridhar, "Offline handwritten Devanagari word recognition: A holistic approach based on directional chain code feature and HMM," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2008, pp. 203–208.

[70] B. Shaw, U. Bhattacharya, and S. K. Parui, "Combination of features for efficient recognition of offline handwritten Devanagari words," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 240–245.

[71] B. Shaw, U. Bhattacharya, and S. K. Parui, "Offline handwritten Devanagari word recognition: Information fusion at feature and classifier levels," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 720–724.

[72] B. Shaw and S. K. Parui, "A two-stage recognition scheme for offline handwritten Devanagari words," in *Machine Interpretation of Patterns: Image Analysis and Data Mining*. Singapore: World Scientific, 2010, pp. 145–165.

[73] B. Singh, A. Mittal, M. Ansari, and D. Ghosh, "Handwritten Devanagari word recognition: A curvelet transform based approach," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 4, pp. 1658–1665, 2011.

[74] S. Kumar, "A study for handwritten Devanagari word recognition," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 1009–1014.

[75] S. K. Parui and B. Shaw, "Offline handwritten Devanagari word recognition: An HMM based approach," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, vol. 4815, 2007, pp. 528–535.

[76] S. Ramachandrula, S. Jain, and H. Ravishankar, "Offline handwritten word recognition in Hindi," in *Proc. Workshop Document Anal. Recognit.*, Dec. 2012, p. 4954.

[77] S. Malakar, P. Sharma, P. K. Singh, M. Das, R. Sarkar, and M. Nasipuri, "A holistic approach for handwritten Hindi word recognition," *Int. J. Comput. Vis. Image Process.*, vol. 7, no. 1, pp. 59–78, Jan. 2017.

[78] H. Kaur and M. Kumar, "Offline handwritten Gurumukhi word recognition using eXtreme gradient boosting methodology," *Soft Comput.*, vol. 25, no. 6, pp. 4451–4464, Mar. 2021.

[79] M. Ghosh, S. Malakar, S. Bhowmik, R. Sarkar, and M. Nasipuri, "Feature selection for handwritten word recognition using memetic algorithm," in *Advances in Intelligent Computing*, vol. 687, J. Mandal, P. Dutta, and S. Mukhopadhyay, Eds. Germany: Springer, 2019, pp. 103–124.

[80] S. Malakar, M. Ghosh, S. Bhowmik, R. Sarkar, and M. Nasipuri, "A GA based hierarchical feature selection approach for handwritten word recognition," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2533–2552, Apr. 2020.
[81] S. Malakar, S. Paul, S. Kundu, S. Bhowmik, R. Sarkar, and M. Nasipuri, "Handwritten word recognition using lottery ticket hypothesis based pruned CNN model: A new benchmark on CMATERdb2.1.2," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 15209–15220, Sep. 2020.
[82] A. K. Bhunia, P. P. Roy, A. Mohta, and U. Pal, "Cross-language framework for word recognition and spotting of indic scripts," *Pattern Recognit.*, vol. 79, pp. 12–31, Jul. 2018.
[83] D. Nurseitov, K. Bostanbekov, M. Kanatov, A. Alimova, A. Abdallah, and G. Abdimanap, "Classification of handwritten names of cities and handwritten text recognition using various deep learning models," 2021, *arXiv:2102.04816*.
[84] R. Bulla, A. K. Gupta, A. Raj, M. Javed, and S. R. Dubey, "HWRCNet: Handwritten word recognition in JPEG compressed domain using CNN-BiLSTM network," 2020, *arXiv:2201.00947*.
[85] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 849–862, Apr. 2018.

NAIWRITA BORAH (Member, IEEE) received the B.Tech. and M.Tech. degrees in information technology from the Sikkim Manipal Institute of Technology (SMIT). She is currently pursuing the Ph.D. degree in computer vision and image processing. She is an Assistant Professor with Jain (Deemed-to-be University). She has made notable contributions to computer vision and image processing as an educator and a researcher. She has written research papers on data engineering, AI in the metaverse, vision transformer-based breast cancer classification, and feature extraction techniques for content-based image retrieval. Her research interests include image processing, computer vision, pattern recognition, and machine learning.

UDAYAN BARUAH received the M.Sc. degree in mathematics from the Faculty of Mathematical Sciences, University of Delhi (North Campus), India, and the M.Tech. degree in information technology and the D.Phil. degree in computer science and engineering from the Department of Computer Science and Engineering, Tezpur (Central) University, Assam, India. He is currently an Associate Professor with the Department of Information Technology, Sikkim Manipal Institute of Technology, Sikkim, India. His primary research interests include online handwriting recognition, document image processing, and medical image processing, among other related areas.

MAHESH THYLORE RAMAKRISHNA (Senior Member, IEEE) is currently an Associate Professor and the Program Head with the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jain (Deemed-to-be University), Bengaluru, India. He has to his credit more than 50 research papers in Scopus/WoS and SCIE indexed journals of high repute. He has been an editor for books on emerging and new age technologies with publishers, such as Springer, IGI Global, and Wiley. He has served as a reviewer and a technical committee member for multiple conferences and journals of high reputation. His research interest include image processing, machine learning, deep learning, artificial intelligence, the IoT, and data science.

V. VINOTH KUMAR (Member, IEEE) is currently an Associate Professor with the School of Information Technology and Engineering, Vellore Institute of Technology University, Tamil Nadu, India. He is the author/coauthor of papers in international journals and conferences, including SCI indexed papers. He has published more than 35 papers in IEEE Access, Springer, Elsevier, IGI Global, and Emerald. His current research interests include wireless networks, the Internet of Things, machine learning, and big data applications. He is currently an Associate Editor of *International Journal of e-Collaboration* (IJeC) and *International Journal of Pervasive Computing and Communications* (IJPCC), and an editorial member of various journals.

D. RAMYA DORAI is currently a Professor with the Department of Computer Engineering, School of Engineering, RK University, Rajkot, Gujarat, India. She has authored more than 20 papers in Scopus/WoS indexed journals. Her research interests include mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs).

JONNAKUTI RAJKUMAR ANNAD received the B.Tech. degree in mechanical engineering from JNT University, Hyderabad, and the M.Tech. degree in mechanical machine design from JNTUK University, India, in 2012. He is currently an Assistant Professor with the Department of Electromechanical Engineering, Sawla Campus, Arba Minch University, Ethiopia. His research interests include mechanical elements design, composite materials, and design analysis of various mechanical parts.