Enhancing Assamese Word Recognition for CBIR: A Comparative Study of Ensemble Methods and Feature Extraction Techniques

Naiwrita Borah*[1], Udayan Baruah[2], Barnali Dey[3], Merin Thomas[4], Sunanda Das[5], Moumi Pandit[6], Bijoyeta Roy[7], Amrita Biswas[8]
PhD Scholar, Department of IT, SMIT, SMU, Sikkim, India[1];
Assistant Professor, Department of CSE, Presidency University, Bangalore, India[1]
Academic Registrar (in-charge) and Controller of Examinations,
Birangana Sati Sadhani Rajyik Vishwavidyalaya (A Government of Assam University), Golaghat, Assam, India[2]
Assistant professor (SG), Department of IT, SMIT, SMU, Sikkim, India[3]
Associate Professor, School of CSE, RV University, Bengaluru, India[4]
Department of CSE, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bengaluru, India[5]
Associate professor, Department of EEE, SMIT, SMU, Sikkim, India[6]
Assistant professor, Department of CSE, SMIT, SMU, Sikkim, India[7]
Department of CSE, SMIT, SMU, Sikkim, India[8]

***Abstract*—This study conducts a thorough assessment of ensemble machine learning methods, specifically focusing on the identification of Assamese words. This task is crucial for improving Content-Based Image Retrieval systems and safeguarding the digital heritage of Assamese culture. We analyze the efficacy of different algorithms, such as CatBoost, XGBoost, Gradient Boosting, Random Forest, Bagging, AdaBoost, Stacking, and Histogram-Based Gradient Boosting, by thoroughly examining their performance in terms of accuracy, precision, recall, Kappa, F1-score, Matthews Correlation Coefficient, and AUC. The CatBoost algorithm stands out as the top performer, achieving an accuracy rate of 97.7%, precision rate of 95%, and recall rate of 96%. XGBoost is also acknowledged for its substantial effectiveness. This comparative analysis emphasizes CatBoost's superiority in terms of precision and recall. Additionally, it underscores the strong ability of ensemble classifiers to enhance assistive technologies, promote social inclusivity, and seamlessly integrate the Assamese language into technological applications.**

Keywords*—Assamese literary works; automatic word recognition; comparative analysis; feature-based approaches; intelligent assistive technology; machine learning; word image analysis

I. Introduction

Assamese, predominantly spoken in the state of Assam and various other regions in Northeast India, holds a significant position as one of the primary languages in India [1]. Hence, the development of a precise Assamese automatic word recognition system holds the potential to safeguard the cultural heritage of India. The recognition of Assamese words is of utmost importance in the preservation and promotion of the Assamese language. An accurate recognition system has the potential to address numerous domains, including digital resource management, the creation of educational tools, and the preservation of digital languages [2]. The ability to accurately recognize words enhances the accessibility of information for individuals who speak Assamese. The utilization of this technology enables the advancement of various technological applications, including content based image retrieval (CBIR), natural language processing (NLP), and information retrieval systems that are specifically tailored for the Assamese language. This facilitates the utilization of digital content, retrieval of pertinent information, and engagement in online platforms by Assamese speakers in their mother tongue.

The Assamese language exhibits dialectal variations, accents, and regional distinctions, which may pose difficulties in word recognition. Ensemble techniques refer to the automated process of effectively identifying and accurately interpreting word images [3]. This is achieved by integrating multiple models or classifiers, thereby leveraging their respective strengths and weaknesses. Ensemble methods have the ability to utilize the combined knowledge and expertise of individual models in order to generate results that are more accurate [4]. The task at hand pertains to the development of computational models, algorithms, and systems with the capability to effectively identify and comprehend Assamese words. In contrast to state-of-the-art methods, ensemble methods exhibit greater accuracy and robustness due to their utilization of multiple models trained on distinct subsets of the data or employing diverse feature representations. In order to facilitate the preservation and expansion of the Assamese language, as well as its integration into society, numerous researchers have employed various methodologies.

A. Distinctive Contribution

This study encompasses the completion of three primary objectives:

1) Identification of handcrafted features for Assamese Word Recognition.
2) Implementation of Ensemble Classification using the handcrafted features.

[*] Corresponding authors

3) Evaluation of performance to determine the optimal ensemble classification method for classification analysis.

The schematic depicted in Fig. 1 offers a graphical illustration of the sequence of tasks involved in this work.

This work is organised in the following format . Section II entails a thorough examination of relevant studies to establish the essential context and background for the research. Section III, entitled “Materials and Methods”, offers a thorough elucidation of the process by which the dataset was generated and the precise methodologies employed for feature engineering. Section IV, of the research paper discusses various ensemble models, while Section V, provides comprehensive details on the performance metrics employed to assess these models. Section VI, presents a comprehensive analysis of the findings, offering a comparative assessment of the effectiveness of the models. In Section VII, the paper concludes by presenting a succinct overview of the primary discoveries and contributions. Section VIII subsequently provides a delineation of prospective paths for future research. In addition, a comprehensive compilation of references is provided to support and validate the research.

II. Related Studies

The authors of [5] used feature extraction techniques such as zoning, chain code, and Fourier descriptors to recognize Assamese handwritten numerals. Table I presents a compilation of the relevant literature pertaining to ensemble methods. The extracted features could be used for word recognition. The study explores a deep learning-based approach for Assamese text recognition [6]. The results show that preprocessing can help a convolutional neural network (CNN) architecture recognize words accurately. Several studies have examined Assamese handwriting recognition issues. An ensemble system uses deep learning models like CNNs and LSTMs to improve recognition accuracy [7]. Ensemble techniques can improve Assamese handwriting recognition and social inclusion, according to this study. The paper proposes an adaptive approach for handwritten Assamese word recognition [8]. The horizontal, vertical, and gradient profiles of word images are used to extract features. The system uses a hybrid classifier that combines SVM and ANN benefits. This study contributes to inclusive word recognition by focusing on Assamese handwriting recognition challenges. This paper focuses on recognizing characters in offline Assamese handwriting [5]. The CNN architecture presented in this study is designed for Assamese character recognition. Word recognition systems require precise character identification, and this study improves Assamese word recognition. This study proposes a Hidden Markov Model (HMM) and Support Vector Machine (SVM) approach for Assamese online character recognition [9]. These models are also compared for efficacy. According to [10], an OCR system for handwritten Assamese characters uses Artificial Neural Networks (ANN) for character segmentation. Character segmentation is done using horizontal and vertical projection on handwritten text. In [11], researchers examine Academic literature proposes algorithms for online handwriting and machine-printed Assamese language text recognition. The Assamese language has more cursive writing than English and others. Feature selection (FS) extracts many features from simple to complex data.

Bagging and AdaBoost classifiers are popular ensemble learning methods for handwritten character recognition. Using different subsets of training data, many researchers have trained decision trees, support vector machines, and neural networks. The authors of [12] show that Bagging and AdaBoost Classifiers enhance devnagari document recognition accuracy. Researchers in [13] proved that statistical features can accurately classify handwritten Assamese language. Several studies have shown that deep learning techniques like CNNs, RNNs, and Deep Boltzmann Machines can effectively recognize handwritten text [14]. The above Assamese word recognition papers offer significant contributions and methods. They demonstrate the ongoing effort to improve social inclusivity by creating accurate and reliable recognition systems. Table IV presents a concise overview of the ensemble learning techniques employed in different scripts.

TABLE I. The Literature Concerning Ensemble Learning Methods Applied to Scripts

Year	Author	Technique	Dataset	Advantages
2007	Sarma[5]	Feature extraction (zoning, chain code, Fourier descriptors)	Handwritten Assamese numerals	Potential for word recognition
2007	Sarma (Character Recognition) [5]	CNN architecture for character recognition	Offline Assamese handwriting	Specific to characters
2013	Sarma[9]	HMM and SVM models	Assamese online characters	Utilizes multiple models
2015	Singh[8]	Hybrid classifier (SVMs, ANNs)	Handwritten Assamese words	Focuses on difficulties
2018	Jangid[14]	Deep learning techniques (CNNs, RNNs, etc.)	Handwritten text	Utilizes deep learning
2019	Alvear[6]	CNN architecture with preprocessing	Assamese text	Utilizes deep learning
2019	Narang[12]	Bagging and AdaBoost Classifiers	Devanagari documents	Ensemble learning
2019	Narang[13]	Statistical features	Handwritten Assamese language	Utilizes statistical features
2021	Choudhury [7]	Ensemble system (CNNs, LSTMs)	Handwritten Assamese text	Societal inclusion
2019	Chourasia [10]	ANN for character segmentation	Handwritten Assamese characters	Specific to character recognition
2022	Ghosh[11]	Various algorithms	Online handwriting and machine-printed Assamese text	Explores various algorithms

TABLE II. Summary of Instance Counts for Each Class Label

Sl. No.	Class Label	Instance Count
1	Burhi_aair_xadhu	2040
2	Bezbaruahr Rasanawali (Vol 2)	2065
3	Kirttana_and_Ghosha	2050
4	Gauburha	2062
5	Jilikoni	2057
	Total	10274

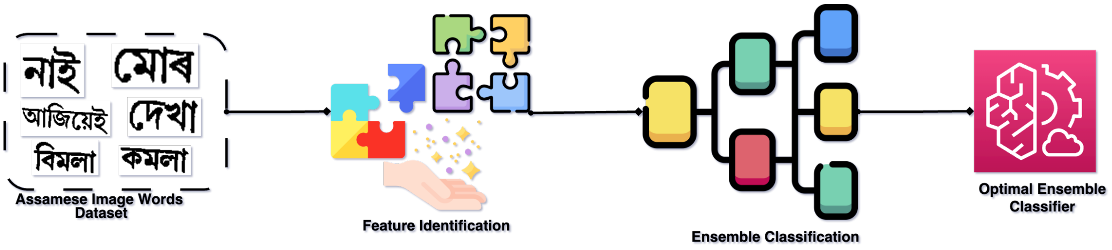


Fig. 1. The workflow of the methodology.

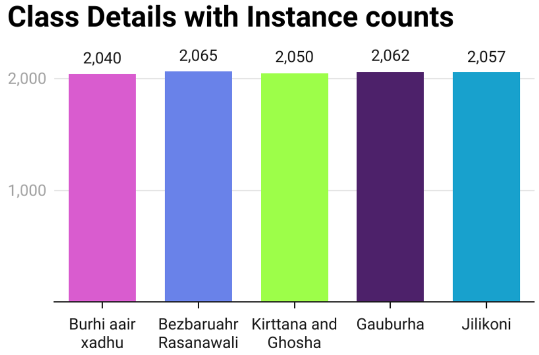


Fig. 2. Class details with instance counts.

III. Materials and Methods

A. Dataset Creation

The dataset curation approach for the experiments entailed a rigorous manual technique. The visual representations were created utilizing publicly accessible web resources related to the Jonaki and Shankari literary periods in Assamese literature. Access to the dataset can be provided upon a formal request, taking into account its potential significance.

The dataset used in this research was carefully chosen and organized. It consisted of photographs obtained from freely available online publications that were related to the Jonaki and Shankari periods in Assamese literature. The importance of studying these literary periods is in their contribution to the cultural heritage of Assam. The collection notably features images from five specific novels, namely, “Burhi Aair Xadhu”, “Bezbaruahr Rasanawali (Vol 2)”, “Kirttana and Ghosha”, “Gauburha”, and “Jilikoni”. The dataset, comprising images extracted from particular books, is presented in Table II, and Fig. 2 displays the specific information regarding the classes and the corresponding number of instances in our dataset.

Borah et al. offer a thorough and complete explanation of a detailed segmentation process [15]. The experimental phase utilized a dataset consisting of 10,274 photographs, which were classified into five unique categories, as detailed in Table II and Fig. 2.

B. Feature Engineering

Table III presents a comprehensive compilation of 1523 distinct characteristics used in the image dataset. These characteristics encompass a wide range of diverse attributes. Every attribute contributes to the creation of a comprehensive representation of handwritten word images. Significant metrics in this context encompass CHA (Convex Hull Area) and CHP (Convex Hull Perimeter), which provide valuable information about the geometric attributes of words. SPLBP (Spatial Layout Binary Pattern) provides valuable information regarding spatial layout attributes, including position, orientation, and scale. The analysis is improved by integrating features such as PHOG (Pyramid of Histograms of Oriented Gradients) and EHD (Edge Histogram Descriptors), which offer insights into texture, structure, and shape. The local shape properties are understood by considering additional characteristics such as kurtosis, rectangularity, volume, compactness, and HuMoments. The recognition system requires a wide variety of features in order to accurately distinguish and categorize handwritten Assamese words. This will improve the accuracy and robustness of ensemble classification methods. Additionally, [16] offers a comprehensive examination of various shape-based features utilized in CBIR.

TABLE III. Features Implemented on the Image Dataset

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

C. Ensemble Classification

Ensemble classification is a commonly employed technique in the domain of machine learning that leverages the combined power of multiple models to enhance the precision of predictions [17]. In order to create an ensemble model that is more reliable and accurate, the proposed methodology integrates forecasts made by numerous base classifiers, also referred to as "weak learners". Ensemble methods, including bagging, boosting, and stacking, are employed to introduce diversity among individual models and collectively mitigate their respective limitations. An illustration of a technique employed in the field of machine learning is bagging, wherein multiple instances of base models are generated on bootstrapped samples. This particular methodology successfully decreases variance and addresses the potential problem of overfitting. By giving weights to samples that were wrongly classified, the boosting algorithm makes it possible for the model's performance to be improved over and over again. The stacking technique entails the amalgamation of multiple models through the utilization of a meta-learner, with the aim of capitalizing on their respective strengths. Ensemble classification is extensively employed across diverse domains, including image recognition, natural language processing, and financial forecasting, with the primary aim of achieving enhanced predictive accuracy. The methodology employed in our study is depicted in Fig. 1, outlining the step-by-step process of improving Assamese word recognition.

1) Gradient Boosting (GB): The technique of Gradient Boosting is employed to optimize a loss function through the sequential addition of weak learners, typically in the form of decision trees [18]. The ultimate forecast is computed by aggregating the individual predictions of the learners using a weighted sum.

Mathematically, in each iteration, we update the model as follows:

$$F_t(x) = F_{t-1}(x) + \arg\min_h \left(\sum_{i=1}^{N} L(y_i, F_{t-1}(x_i) + h(x_i)) \right) \tag{1}$$

Definitions:

1. $F_t(x)$: The ensemble's prediction at iteration t. 2. $h(x)$: The weak learner's prediction. 3. $L(y_i, F(x_i))$: The loss function, typically squared error for regression or cross-entropy for classification.

2) CatBoost (CB): In order to enhance the training procedure, CatBoost employs the ordered boosting technique, which takes into account the ordering of categorical variables [19]. This methodology facilitates the preservation of the inherent hierarchy of categorical attributes, which can prove advantageous in a multitude of contexts including recommendation and ranking systems. By integrating the natural order of categorical variables, CatBoost has the capability to augment the model's predictive performance.

CatBoost, apart from employing ordered boosting, incorporates a statistical technique to mitigate the risk of overfitting in the context of categorical data. Overfitting occurs when a model learns an excessive amount of the training data, including any noise or random fluctuations, which can result in inadequate generalization to new data. CatBoost implements a regularization technique to mitigate the risk of overfitting, specifically in the context of categorical features.

From a mathematical standpoint, it can be observed that this approach optimizes the identical loss function as gradient boosting. However, it distinguishes itself by employing distinct methods to handle categorical features.

$$\text{CB} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{j=1}^{J} \Omega(C_j) \tag{2}$$

Where: - $L(y_i, F(x_i))$ is the loss function that measures the difference between the predicted values and the true labels.
- $F(x_i)$ represents the model's prediction for the i-th data point.
- C_j represents categorical features and $\Omega(C_j)$ is a regularization term applied specifically to categorical features.
- J is the total number of categorical features.

3) Random Forest (RF): Random Forest is a machine learning algorithm that operates by aggregating multiple decision trees. As such, it does not possess a singular equation that fully encompasses its functionality. The RF algorithm creates a collection of decision trees and aggregates their predictions using either voting (for classification [20], [21]) or averaging (for regression [22]).

4) XGBoost: XGBoost effectively optimizes the given objective function in order to construct an ensemble of decision trees, rendering it a robust and efficient algorithm that is extensively employed in machine learning competitions.

$$\text{XGBoost} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{k=1}^{K} \Omega(f_k) \tag{3}$$

Where:
-L(y_i, F(x_i)) & is the loss function.
-Omega(f_k) & is the regularization term for each tree.

5) Bagging: Bagging improves decision tree-based predictive models' accuracy and resilience. Original model is a "weak learner", usually a decision tree. Bootstrap samples create multiple instances by training the base model on different training data subsets. This subset is created by random sampling with replacement. Academic literature calls these subsets "bootstrap samples". Multiple base models trained on bootstrap samples are exposed to slightly different dataset variations. Predicting future events. Model predictions are aggregated. Regression aggregation averages predictions, while classification determines majority vote. Bagging reduces variance, overfitting, and generalization by using trained model heterogeneity. Thus, it enhances prediction.

The Bagging prediction can be represented as:

$$\text{Bagging Prediction} = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{4}$$

Where: - Bagging Prediction is the final prediction made by Bagging. - N is the number of base models (often decision trees) created through bootstrapped samples. - $f_i(x)$ represents the prediction of the i-th base model on input data x.

6) AdaBoost: AdaBoost, also known as Adaptive Boosting, is a machine learning algorithm that employs the technique of ensemble learning to construct a robust model by aggregating multiple weak learners [23]. The algorithm employs an iterative process whereby distinct weights are assigned to each weak learner in accordance with their respective performance. The primary objective is to rectify the errors made by preceding models. The ultimate forecast is derived by calculating a weighted sum of the prognostications made by these inferior learners.

The AdaBoost algorithm can be mathematically summarized as follows:

1) **Initialization:** Start by initializing the sample weights w_i uniformly, where i ranges from 1 to the number of training samples.
2) **Iteration t:**
    a) Train a weak learner, denoted as $h_t(x)$, on the training data with the current sample weights w_i.
    b) Compute the weighted error ϵ_t of $h_t(x)$ on the training data:

$$\epsilon_t = \sum_{i=1}^{N} w_i \cdot I(y_i \neq h_t(x_i)) \tag{5}$$

where N is the number of training samples, y_i is the true label, x_i is the input data, and I is the indicator function.

    c) Compute the importance weight of $h_t(x)$:

$$\alpha_t = \frac{1}{2} \cdot \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{6}$$

    d) Update the sample weights for the next iteration:

$$w_{i,t+1} = w_i \cdot \exp\left(-\alpha_t \cdot y_i \cdot h_t(x_i)\right) \tag{7}$$

Normalize the weights so that they sum up to 1:

$$w_{i,t+1} = \frac{w_{i,t+1}}{\sum_{i=1}^{N} w_{i,t+1}} \tag{8}$$

3) Repeat the above steps for a predefined number of iterations or until a stopping criterion is met.
4) The final prediction $F(x)$ is obtained by combining the predictions of weak learners with their importance weights:

$$F(x) = \sum_{t=1}^{T} \alpha_t \cdot h_t(x) \tag{9}$$

7) Stacking: Stacking combines multiple base models by training a meta-model on their predictions. The meta-model learns to weigh the predictions of the base models optimally [25].

Mathematically, stacking can be represented as follows:

$$\hat{y} = g(f_1(x), f_2(x), \ldots, f_k(x)) \tag{10}$$

Where:

$\hat{y}$ is the final prediction.

$f_i(x)$ are the predictions of individual base models.

g is the meta-model, which can be a LR, DT etc.

8) Histogram-based Gradient Boosting: Histogram-Based Gradient Boosting uses histograms to speed up training [24]. It optimizes the same loss function as traditional gradient boosting but employs histogram-based techniques for better efficiency.

The mathematical details of HistGradientBoosting involve optimizing the loss function similar to gradient boosting but with histogram-specific optimizations.

The optimization objective of Histogram-Based Gradient Boosting (HistGradientBoosting) can be summarized as follows:

$$\min_{F(x)} \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{j=1}^{J} \Omega(C_j) \tag{11}$$

Where: - $\min_{F(x)}$ denotes the minimization of the objective with respect to the ensemble model $F(x)$. - N is the number of training samples. - $L(y_i, F(x_i))$ is the loss function that measures the difference between the true labels y_i and the predictions $F(x_i)$. - J represents the categorical features, and $\Omega(C_j)$ is a regularization term specific to categorical features.

This equation provides a simplified representation of the optimization objective in HistGradientBoosting, highlighting the key components involved in gradient boosting with histogram-based techniques.

TABLE IV. Summary of Ensemble Models Assamese Word Recognition

Model	Description
Gradient Boosting (GB)	Gradient Boosting is an ensemble learning method that builds multiple decision trees sequentially. Each tree corrects the errors of the previous one. It's a powerful model for classification and regression tasks, known for its high predictive accuracy.
CatBoost (CB)	CatBoost is a gradient boosting algorithm that is particularly effective for categorical feature handling. It automatically handles categorical data, reducing the need for preprocessing, and can work well with both numerical and categorical features.
Random Forest (RF)	Random Forest is an ensemble of decision trees. It builds multiple trees and combines their predictions through voting (classification) or averaging (regression). It's robust, handles high-dimensional data well, and is less prone to overfitting.
XGBoost	XGBoost (Extreme Gradient Boosting) is an efficient gradient boosting algorithm known for its speed and performance. It's highly customizable and widely used in machine learning competitions due to its predictive power.
Bagging	Bagging stands for Bootstrap Aggregating. It's an ensemble method that builds multiple instances of a base model (usually decision trees) on bootstrapped samples of the data. It reduces variance and helps in avoiding overfitting.
AdaBoost	AdaBoost (Adaptive Boosting) is another boosting algorithm that focuses on the weaknesses of the base model. It assigns weights to misclassified samples and combines multiple weak learners to create a strong ensemble model.
Stacking	Stacking, or Stacked Generalization, combines multiple base models by training a meta-model on their predictions. It leverages the strengths of different models, potentially improving overall performance.
HistGradientBoosting	Histogram-Based Gradient Boosting is an efficient variant of gradient boosting that uses histograms to speed up training. It's particularly useful for large datasets and high-dimensional data.

IV. Performance Metrics

In the domain of machine learning evaluation, a variety of fundamental performance metrics are frequently utilized to

TABLE V. Performance Metrics for Various Machine Learning Models

Model	Accuracy	Precision	Recall	Kappa	F1-score	MCC	Build Time (s)	Run Time (s)	AUC
GB	0.9603	0.9603	0.9603	0.9504	0.9603	0.9504	142.4156	0.0255	0.9972
CB	0.9770	0.9771	0.9770	0.9713	0.9770	0.9713	27.2524	0.5145	0.9984
RF	0.9366	0.9371	0.9366	0.9207	0.9366	0.9208	2.2948	0.0403	0.9943
XGBoost	0.9751	0.9751	0.9751	0.9689	0.9751	0.9689	15.5116	0.0125	0.9989
Bagging	0.8828	0.8829	0.8828	0.8535	0.8827	0.8536	8.0071	0.0876	0.9786
AdaBoost	0.7777	0.7831	0.7777	0.7222	0.7785	0.7233	6.0609	0.1455	0.9271
Stacking	0.9786	0.9787	0.9786	0.9732	0.9786	0.9733	796.6909	4.9693	0.9988
HistGB	0.9743	0.9744	0.9743	0.9679	0.9743	0.9679	36.7397	0.0888	0.9990

TABLE VI. EXISTING VS PROPOSED FEATURE-BASED METHODS

Authors	Scripts	Word Count	Feature Set	Classifier	Accuracy (%)
Shaw and Parui [26]	Devanagari	13,000	Stroke based (Stage-1); Wavelet (Stage-2)	HMM (Stage-1);	91.25
Singh et al. [27]	Devanagari	28,500	Curvelet transform	SVM and KNN	85.6 (SVM); 93.21 (KNN)
Singh[28]	Devanagari	20,000	Combination of uniform zoning, diagonal and centroid features	Gradient boosted decision tree	94.33
Malakar et al. [29]	Hindi	4,620	Low-level features	MLP	96.82
Kaur and Kumar [30]	Gurumukhi	40,000	Zoning features	XGBoost	91.66
Ghosh et al. [31]	Bangla	7,500	Gradient features and modified SCF; MA-based wrapper filter selection approach	MLP	93
Malakar et al.[32], [33]	Bangla	12,000	Gradient-based and elliptical	MLP	95.3

Bhunia et al. [34]	Bangla, Devanagari, Gurumukhi	3,856;
		3,589;
		3,142
			PHOG feature	HMM (Middle-zone), SVM (Upper/Lower zone)	>60

Proposed method (Feature-Based Ensemble)	Assamese	10274	Combination of Multiple Low level - Shape, Region, Descriptor-Based Features,	CatBoost, XGBoost	97.7 (CB), 96.0 (XGBoost)

evaluate the efficacy of classification models. These metrics offer significant insights into the predictive capabilities and overall quality of a model. In this section, we present a concise summary of several key performance metrics.

A. Accuracy

Accuracy is a crucial metric that quantifies the ratio of accurately classified instances to the total number of instances in the dataset. The aforementioned statement offers a comprehensive evaluation of the accuracy of the model's predictions on a global scale.

B. Precision

Precision is a metric that quantifies the ratio of correctly predicted positive instances to the total number of positive predictions made by the model. The evaluation assesses the model's capacity to minimize the occurrence of false positive predictions.

C. Recall

The term “recall”, which is also referred to as sensitivity or true positive rate, is a metric used to measure the proportion of true positive predictions in relation to all the actual positive instances. The evaluation measures the model's capacity to accurately identify and include all pertinent positive instances.

D. Kappa

The Cohen's Kappa statistic serves as a quantitative measure to assess the level of agreement between the predictions generated by a model and the observed outcomes. The method takes into consideration the potential occurrence of coincidental agreement and offers an indication of the model's efficacy beyond what would be expected by random chance.

E. F1-score

The F1-score can be defined as the mathematical average of precision and recall, specifically calculated using the harmonic mean. The provided analysis presents a comprehensive evaluation of the model's capacity to effectively balance precision and recall, making it particularly advantageous in scenarios involving imbalanced datasets.

F. Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) is a metric used to evaluate the degree of correlation between the predictions made by a model and the actual labels assigned to the data. It takes into account all four categories: true positives, true negatives, false positives, and false negatives. The aforementioned metric offers a thorough evaluation of the performance of the model.

G. Build Time (s) and Run Time (s)

The term “Build Time” denotes the duration, measured in seconds, necessary for the training or construction of a machine learning model. The metric of Run Time quantifies the duration, expressed in seconds, required for generating predictions on novel and unobserved data. The metrics presented in this context serve as indicators of the computational efficiency of the model.

H. AUC Area Under the ROC Curve (AUC)

The quantification of a model's ability to differentiate between positive and negative classes is accomplished by the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC). A greater Area Under the Curve (AUC) value signifies an enhanced capacity to differentiate between different classes.

Performance metrics play a crucial role in the assessment and comparison of the efficacy of machine learning models in classification tasks.

I. Training, Testing and Validation Details

The dataset was divided into training and testing sets using a 70-30 split, allowing for a comprehensive evaluation of ensemble classification techniques for identifying Assamese words. More precisely, 70% of the data was allocated for training and denoted by the variables X_{train} and y_{train}. The remaining 30% of the data, referred to as X_{test} and y_{test}, was used for testing. The division was performed using the `train_test_split` function, with the `test_size` parameter set to 0.3. In addition, an additional 20% of the training set was allocated for validation purposes. The validation subset played a vital role in optimizing and validating the models. It enabled the adjustment of parameters and the prevention of overfitting, ensuring the best possible performance of the model on new, unseen data. The results were greatly impacted by the implementation of the data segmentation strategy, which offered a thorough approach for training and evaluating the model. This, in turn, improved the reliability and accuracy of the findings.

J. Experimental Environment Details

The study was conducted on a computational system in the experimental environment, which had the following specifications. The system utilized Python version 3.8.17 and functioned on the Darwin operating system with Kernel Version 23.2.0. The system architecture was 64-bit, with a RAM capacity of 32.00 GB. The CPU was equipped with a configuration consisting of 10 cores and 10 threads. The hardware and software specifications enhance the transparency and reproducibility of the experimental setup, guaranteeing a strong basis for the study's results.

V. Results

The present study provides a comprehensive evaluation of various machine learning models within the framework of a classification task. Each model is assessed using a variety of performance metrics, including accuracy, precision, recall, Kappa, F1-score, MCC (Matthews Correlation Coefficient), build time, run time, and AUC (Area Under the ROC Curve). Testing how well a model works involves checking how well it correctly labels instances, how well it balances precision and recall, how well it matches real-world results, and how quickly it can do the calculations (see Table V). It is noteworthy to mention that specific models, such as CatBoost (CB) and XGBoost, demonstrate a significant degree of accuracy and AUC values, indicating their strong discriminatory abilities. In contrast, AdaBoost demonstrates reduced accuracy and Matthews Correlation Coefficient (MCC) values, suggesting the existence of potential areas for improvement as seen in Fig. 3 and 4a. The table serves as a valuable instrument for selecting the most suitable machine learning model based on specific classification requirements, considering both predictive accuracy and computational efficiency. Fig. 3 illustrates the performance of the classifiers on our dataset, specifically emphasizing accuracy and precision and Fig. 4a and 4b provides a performance analysis of the models, showcasing their ROC values and computational efficiency.

VI. Discussion

This study offers a comprehensive assessment of machine learning models in the context of a classification task. This analysis yields significant insights of note.

The CatBoost (**CB**) model demonstrates superior performance across various metrics, establishing its dominance in the field. The model demonstrates exceptional performance in terms of accuracy (97.7%), precision, and recall, highlighting its proficiency in making precise predictions and effectively capturing positive instances.

The competence of Gradient Boosting (**GB**) is notable, as it consistently demonstrates high accuracy (96.0%) and performs competitively across multiple metrics. It is a highly suitable option for precise classification tasks.

The achievement of high F1-scores, which effectively balance precision and recall, is notable in the performance of **CB** and **XGBoost**. This particular attribute holds significant value when addressing datasets that exhibit imbalances.

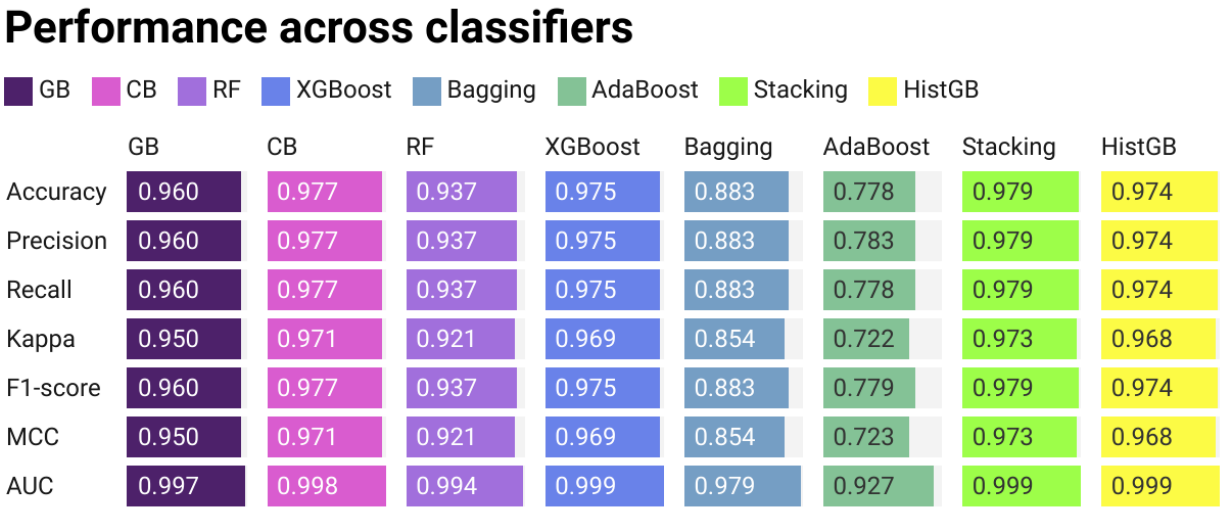
The observed Kappa values for **CB** and **Stacking** indicate a significant level of agreement that surpasses what would be expected by chance alone. This suggests their credibility in generating forecasts that surpass mere chance concurrence.

The discriminatory power of all models is exceptional, as evidenced by their AUC values that approach 1. This suggests their proficiency in effectively discerning between positive and negative classes.

Computational efficiency is a crucial aspect to consider when evaluating the performance of algorithms such as **CB** and **XGBoost**. Although these algorithms exhibit exceptional performance, it is important to note that they necessitate a greater allocation of computational resources. In contrast, **AdaBoost** and **Random Forest** algorithms provide expedited construction and execution durations, rendering them appropriate for situations where computational efficiency is of paramount importance.

A. Perfomance with Reference to other Script based Works

The data as seen in Table VI is a comparative analysis of the limited number of techniques utilized in the field of document recognition and analysis across a variety of scripts. The methodologies being evaluated are attributed to distinguished researchers who have implemented unique approaches for the extraction and categorization of features. The comparative analysis in question examines a wide range of scripts, which comprise Assamese, Devanagari, Hindi, Gurumukhi, and Bangla.

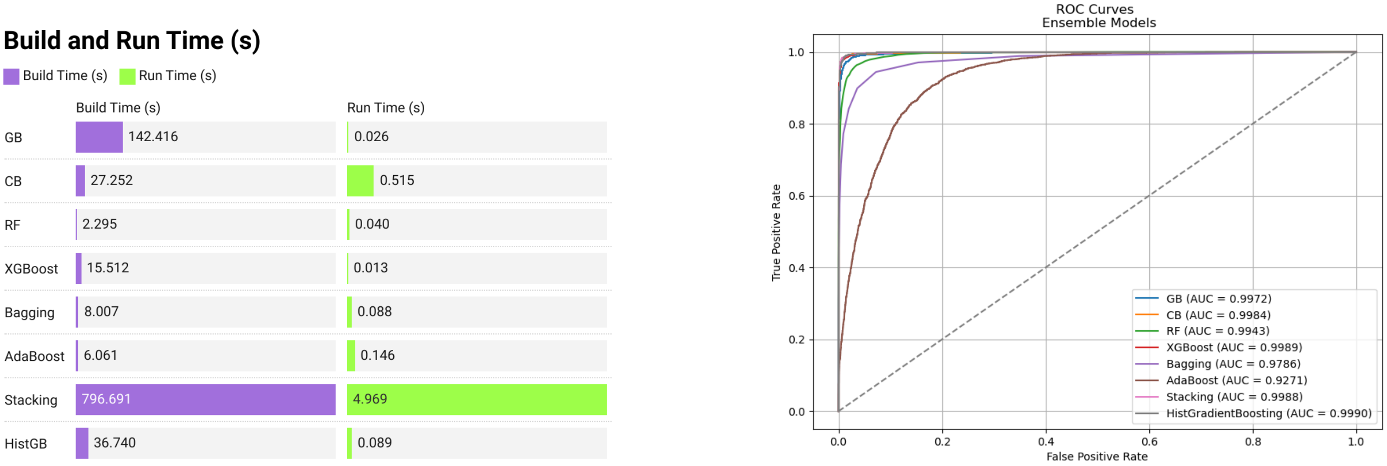


Performance across classifiers

🟪 GB 🟪 CB 🟪 RF 🟦 XGBoost 🟦 Bagging 🟩 AdaBoost 🟩 Stacking 🟨 HistGB

	GB	CB	RF	XGBoost	Bagging	AdaBoost	Stacking	HistGB
Accuracy	0.960	0.977	0.937	0.975	0.883	0.778	0.979	0.974
Precision	0.960	0.977	0.937	0.975	0.883	0.783	0.979	0.974
Recall	0.960	0.977	0.937	0.975	0.883	0.778	0.979	0.974
Kappa	0.950	0.971	0.921	0.969	0.854	0.722	0.973	0.968
F1-score	0.960	0.977	0.937	0.975	0.883	0.779	0.979	0.974
MCC	0.950	0.971	0.921	0.969	0.854	0.723	0.973	0.968
AUC	0.997	0.998	0.994	0.999	0.979	0.927	0.999	0.999

Fig. 3. The performance of several classifiers on the dataset.



(a) Build and run times of several classifiers.

(b) ROC values for classifiers.

Fig. 4. Performance analysis of models.

The methods described herein employ datasets with substantial variation in word count, which spans from 3,142 to 40,000. This discrepancy is indicative of the vast and varied scale of the document corpora under investigation. Prominent feature extraction methodologies include modified spatial co-occurrence functions (SCF), stroke-based techniques, curvelet transforms, combinations of zoning and centroid features, low-level features, zoning features, and gradient features. Moreover, the integration of ensemble techniques, exemplified by the proposed method, encompasses a multitude of low-level characteristics, including those based on shape, region, and descriptors.

Hidden Markov Models (HMM), Support Vector Machines (SVM), k-Nearest Neighbors (KNN), gradient-boosted decision trees, Multi-Layer Perceptrons (MLP), XGBoost, and CatBoost are among the classifiers utilized in the assessed methodologies. The aforementioned classifiers demonstrate an extensive array of algorithmic methodologies, which accurately represents the intricate demands of document analysis across various scripts.

One crucial metric for evaluating the effectiveness of the suggested methodologies is accuracy. The range of obtained accuracy values is between 85.6% and 97.7%. The performance of the proposed method, which employs a feature-based ensemble approach for Assamese script recognition, is noteworthy. By attaining an accuracy of 97.7% with CatBoost and 96.0% with XGBoost, this ensemble method solidifies its position as a formidable competitor within the realm of script recognition methodologies.

VII. Conclusion

The investigation of Assamese word recognition demonstrated superior performance in terms of F1-scores, accuracy, precision, and recall. This was achieved by utilizing ensemble methods and feature extraction techniques, with a specific focus on the effectiveness of CatBoost and XGBoost.

The research inquiry has emphasized the findings regarding the efficacy of different ensemble methods, particularly in the identification of Assamese words. Furthermore, it is emphasized that although AdaBoost and Random Forest can be effective alternatives, especially in scenarios with limited computational resources, they demonstrate slightly lower performance metrics compared to CatBoost and XGBoost.

The methodology's resilience and practicality are showcased by employing a comprehensive dataset comprising 10,000 words and diverse feature extraction methodologies.

A significant advancement in the field of computational linguistics has been achieved by successfully developing a method for recognizing Assamese words. This has resulted in the promotion of technological diversity across different languages.

VIII. Future Work

This study presents various opportunities for future research. An important focus is the incorporation of deep learning methods to improve the process of extracting distinctive characteristics and accurately categorizing Assamese words. Investigating recurrent neural networks and convolutional neural networks has the potential to yield substantial enhancements. Moreover, augmenting the dataset to encompass a wider range of handwriting styles and integrating multi-script recognition systems would significantly bolster the model's resilience. Finally, exploring the practical and influential application of these models in real-time on mobile devices and web applications would be a worthwhile direction.

IX. Conflict of Interest

The authors declare that there is no conflict of interest.

References

[1] S. Mahanta, *Assamese*, *Journal of the International Phonetic Association*, vol. 42, no. 2, pp. 217–224, 2012, doi: 10.1017/s0025100312000096.

[2] G. Upadhye, U. Kulkarni, and D. Mane, *Improved model configuration strategies for Kannada handwritten numeral recognition*, *Image Analysis & Stereology*, vol. 40, no. 3, pp. 181-191, 2021, doi: 10.5566/ias.2586.

[3] Y. Wen and R. Filik, *Electrophysiological dynamics of Chinese phonology during visual word recognition in Chinese-English bilinguals*, *Scientific Reports*, vol. 8, no. 1, 2018, doi: 10.1038/s41598-018-25072-w.

[4] S. Yuan, Y. Wei, and D. Zhao, *Computer-aided lung nodule recognition by SVM classifier based on combination of random undersampling and SMOTE*, *Computational and Mathematical Methods in Medicine*, vol. 2015, pp. 1-13, 2015, doi: 10.1155/2015/368674.

[5] K. K. Sarma, *MLP-based Assamese Character and Numeral Recognition using an Innovative Hybrid Feature Set*, in *IICAI*, December 2007, pp. 585-600, doi: 10.1109/IICAI.2007.357.

[6] R. F. Alvear-Sandoval, J. L. Sancho-Gómez, and A. R. Figueiras-Vidal, *On improving CNNs performance: The case of MNIST*, *Information Fusion*, vol. 52, pp. 106-109, 2019, doi: 10.1016/j.inffus.2019.03.016.

[7] A. Choudhury and K. K. Sarma, *A CNN-LSTM based ensemble framework for in-air handwritten Assamese character recognition*, *Multimedia Tools and Applications*, pp. 1-36, 2021, doi: 10.1007/s11042-021-11572-4.

[8] P. K. Singh, R. Sarkar, and M. Nasipuri, *Word-Level Script Identification Using Texture Based Features*, *International Journal of System Dynamics Applications (IJSDA)*, vol. 4, no. 2, pp. 74-94, 2015, doi: 10.4018/ijsda.2015040105.

[9] B. Sarma, K. Mehrotra, R. K. Naik, S. Raj Prasanna, S. Belhe, and C. Mahanta, *Handwritten Assamese Numeral Recognizer using HMM & SVM Classifiers*, in *2013 National Conference on Communications (NCC)*, February 2013, pp. 1-5, doi: 10.1109/NCC.2013.6487976.

[10] C. K. Chourasia and M. Barman, *Handwritten Assamese Character Recognition*, in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, March 2019, pp. 1-6, doi: 10.1109/I2CT45656.2019.9033947.

[11] T. Ghosh, S. Sen, S. M. Obaidullah, K. C. Santosh, K. Roy, and U. Pal, *Advances in Online Handwritten Recognition in the Last Decades*, *Computer Science Review*, vol. 46, 2022, pp. 100515, doi: 10.1016/j.cosrev.2022.100515.

[12] S. R. Narang, M. K. Jindal, and M. Kumar, *Devanagari Ancient Character Recognition Using DCT Features with Adaptive Boosting and Bootstrap Aggregating*, *Soft Computing*, vol. 23, 2019, pp. 13603-13614, doi: 10.1007/s00500-019-03973-2.

[13] S. Narang, M. K. Jindal, and M. Kumar, *Devanagari Ancient Documents Recognition Using Statistical Feature Extraction Techniques*, *Sādhanā*, vol. 44, 2019, pp. 1-8, doi: 10.1007/s12046-018-0997-7.

[14] M. Jangid and S. Srivastava, *Handwritten Devanagari Character Recognition Using Layer-Wise Training of Deep Convolutional Neural Networks and Adaptive Gradient Methods*, *Journal of Imaging*, vol. 4, no. 2, 2018, pp. 41, doi: 10.3390/jimaging4020041.

[15] N. Borah, U. Baruah, T. R. Mahesh, V. V. Kumar, D. R. Dorai, and J. Rajkumar Annad, *Efficient Assamese Word Recognition for Societal Empowerment: A Comparative Feature-Based Analysis*, *IEEE Access*, vol. 11, 2023, pp. 82302-82326, doi: 10.1109/ACCESS.2023.3301564.

[16] N. Borah and U. Baruah, *Feature Extraction Techniques for Shape-Based CBIR—A Survey*, in *Contemporary Issues in Communication, Cloud and Big Data Analytics*, H. K. D. Sarma, V. E. Balas, B. Bhuyan, and N. Dutta (Eds.), Lecture Notes in Networks and Systems, vol. 281, Springer, 2022, pp. 205-214, doi: 10.1007/978-981-16-4244-9_16.

[17] S. Priya, A. Agarwal, C. Ward, T. Locke, V. Monga, and G. Bathla, "Survival Prediction in Glioblastoma on Post-Contrast Magnetic Resonance Imaging Using Filtration-Based First-Order Texture Analysis: Comparison of Multiple Machine Learning Models," *The Neuroradiology Journal*, vol. 34, no. 4, pp. 355-362, 2021. doi: 10.1177/1971400921990766

[18] B. Fernandes, A. González-Briones, P. Novais, M. Calafate, C. Analide, and J. Neves, "An Adjective Selection Personality Assessment Method Using Gradient Boosting Machine Learning," *Processes*, vol. 8, no. 5, p. 618, 2020. doi: 10.3390/pr8050618

[19] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," arXiv preprint arXiv:1706.09516, 2017. doi: 10.48550/arxiv.1706.09516

[20] B. Balnarsaiah, T. Prasad, and P. Laxminarayana, "Pixel-Based SAR Image Classification Using Random Forest Algorithm," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 4351-4356, 2019. doi: 10.35940/ijitee.j9873.0881019

[21] D. Chutia, N. Borah, D. Baruah, et al., *An Effective Approach for Improving the Accuracy of a Random Forest Classifier in the Classification of Hyperion Data*, *Applied Geomat*, vol. 12, pp. 95–105, 2020, doi: 10.1007/s12518-019-00281-8.

[22] A. Jog, A. Carass, S. Roy, D. Pham, and J. Prince, "Random Forest Regression for Magnetic Resonance Image Synthesis," *Medical Image Analysis*, vol. 35, pp. 475-488, 2017. doi: 10.1016/j.media.2016.08.009

[23] A. Lykov, S. Muzychka, and K. Vaninsky, "The AdaBoost Flow," *Communications on Pure and Applied Mathematics*, vol. 68, no. 5, pp. 865-886, 2014. doi: 10.1002/cpa.21555

[24] M. Kashifi and I. Ahmad, "Efficient Histogram-Based Gradient Boosting Approach for Accident Severity Prediction with Multisource Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2676, no. 6, pp. 236-258, 2022. doi: 10.1177/03611981221074370

[25] O. Petinrin and F. Saeed, "Stacked Ensemble for Bioactive Molecule Prediction," *IEEE Access*, vol. 7, pp. 153952-153957, 2019. doi: 10.1109/access.2019.2945422

[26] 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, pp. 145-165, World Scientific, 2010.

[27] B. Singh, A. Mittal, M. Ansari, and D. Ghosh, ”Handwritten Devanagari word recognition: a curvelet transform based approach,” *International Journal of Computer Science and Engineering*, vol. 3, no. 4, pp. 1658-1665, 2011.

[28] S. Singh, N. K. Garg, and M. Kumar, ”On the performance analysis of various features and classifiers for handwritten Devanagari word recognition,” *Neural Computing and Applications*, vol. 35, no. 10, pp. 7509-7527, 2023.

[29] S. Malakar, P. Sharma, P.K. Singh, M. Das, R. Sarkar, and M. Nasipuri, ”A holistic approach for handwritten Hindi word recognition,” *International Journal of Computer Vision and Image Processing (IJCVIP)*, vol. 7, no. 1, pp. 59-78, 2017.

[30] H. Kaur and M. Kumar, ”Offline handwritten Gurumukhi word recognition using eXtreme gradient boosting methodology,” *Soft Computing*, vol. 25, no. 6, pp. 4451-4464, 2021.

[31] M. Ghosh, S. Malakar, S. Bhowmik, R. Sarkar, and M. Nasipuri, ”Feature selection for handwritten word recognition using memetic algorithm,” In: J. Mandal, P. Dutta, and S. Mukhopadhyay (eds), *Advances in Intelligent Computing*, Studies in Computational Intelligence, vol. 687, pp. 103-124, 2019.

[32] S. Malakar, M. Ghosh, S. Bhowmik, R. Sarkar, and M. Nasipuri, ”A GA based hierarchical feature selection approach for handwritten word recognition,” *Neural Computing and Applications*, vol. 32, no. 7, pp. 2533-2552, 2020.

[33] 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 CMATERdb212,” *Neural Computing and Applications*, vol. 32, no. 18, pp. 15209-15220, 2020.

[34] A. K. Bhunia, P. P. Roy, A. Mohta, and U. Pal, ”Cross-language framework for word recognition and spotting of Indic scripts,” *Pattern Recognition*, vol. 79, pp. 12-31, 2018