"Start As If You Know Nothing"

- Krishnamurti

Chapter 7

RECOGNITION OF OFFLINE HANDWRITTEN DEVANAGARI WORDS USING GRADIENT BOOSTED DECISION THREE (GBDT) APPROACH

7.1 INTRODUCTION

This chapter focuses on the application of the holistic approach, also known as segmentation-free, for the recognition of offline handwritten Devanagari words. Unlike the analytic approach, which analyzes individual characters or components, the holistic approach recognizes the entire word image as a single entity. This approach proves to be more effective, especially for small lexicon sizes. The chapter explores the use of statistical features to generate feature vector sets that describe each word in the feature space. These features include uniform zoning, diagonal and centroid-based features extracted from a database of handwritten word images consisting of 50-word classes. Various classifiers such as K-Nearest Neighbor (KNN), Decision Tree (DT) and Random Forest (RF) are utilized for the recognition task. Additionally, to improve the system's performance, a combination of the aforementioned features along with the Gradient Boosted Decision Tree (GBDT) algorithm is proposed.

The chapter is divided into six sections, each addressing different aspects of the topic. Section 7.2 provides an explanation of the feature extraction techniques, emphasizing their relevance to the study. In Section 7.3, the Gradient Boosted Decision Tree (GBDT) approach is discussed as a means to enhance the system's performance. The experimental results, focusing on the selected features and the application of the GBDT approach, are presented in Section 7.4. Section 7.5 offers a comparative analysis, comparing the findings of the present work with other relevant studies. Finally, in

Section 7.6, the entire chapter is summarized, highlighting the key points and conclusions.

7.2 FEATURE EXTRACTION TECHNIQUES

The purpose of feature extraction techniques are to capture the relevant shape information of the character/word, leading to a higher recognition rate. The quality of the extracted features greatly influences the overall performance of the recognition system. However, extracting effective features can be challenging due to the inherent variability and roughness in handwriting. In optical character/word recognition applications, it is important to select features that can effectively differentiate between different character/word classes within the recognition system. This work focuses on three statistical feature extraction techniques: uniform zoning features, diagonal features and centroid features. The performance of various combinations of these techniques is analyzed in this study.

7.2.1 Uniform Zoning-based Features

The uniform zoning feature extraction method involves dividing a handwritten scanned image into equal-sized zones ($n = 64$), as shown in Fig. 7.1. Each zone is then analyzed to count the number of foreground pixels ($p_1, p_2, \dots \dots .., p_n$) present within it. These counts are normalized to a range of $[0, 1]$, resulting in a feature set of n elements (Kaur and Kumar, 2021b).

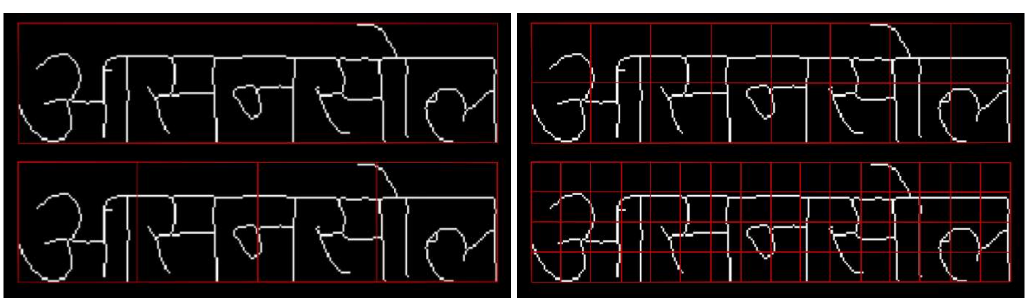

Figure 7.1: Various uniform zones of a word "आसनसोल" (Asansol) written in Devanagari script: **(a)** A single uniform zone considering the whole word image **(b)** 1×4 zones of a word image **(c)** 2×8 zones of a word image and **(d)** 4×16 zones of a word image

In this study, a scanned handwritten word image of size 256×64 (refer Fig. 7.1a) is divided into four equal zones (refer Fig. 7.1b). Each of these zones is then subdivided into 16 smaller zones (refer Fig. 7.1c). Furthermore, each block within these 16 zones is divided into four additional zones (refer Fig. 7.1d). As a result, a total of 85 $(1 + 4 + 16 + 64)$ zones are utilized to calculate the foreground pixel density of the handwritten word image.

Steps involved to extract uniform zoning features are as follows:

Step 1: The first step in the process is to take a digitized image of a word as input and convert it into a binary image. This involves transforming the image so that each pixel is either black (representing the foreground) or white (representing the background). This binary representation allows for easier analysis and extraction of features from the word image.
Step 2: Divide the bitmapped word image into n number of equal-sized zones in a hierarchical order. This zoning process helps to partition the word image into manageable sections, allowing for localized analysis and feature extraction.
Step 3: Count black pixels (foreground pixels) in each zone and calculate density of the zone as given in Eq. 7.1.

$$\text{Density of the zone} = \frac{\Delta}{\in} \tag{7.1}$$

Where, Δ represents total number of foreground pixels (black pixels) in a zone); and $\in$ represents total number of pixels in a zone.
Step 4: Store the density value of each zone into a feature vector. By storing these density values in a feature vector, the system can capture the spatial distribution of the word image and use it for further analysis and recognition purposes.

Finally, 85 features of a handwritten word image are extracted.

7.2.2 Diagonal-based Features

Diagonal based features have been used to play a significant role in achieving higher recognition accuracy (Pradeep et al., 2010; Kaur and Kumar, 2021b). In this method, the scanned handwritten word image is divided into equal-sized zones. Subsequently, features are extracted by moving along the diagonals of the image, as illustrated in Fig. 7.2. The feature vector is calculated on the basis on pixels around each diagonal. This approach allows for capturing important spatial information along the diagonals, which can contribute to improving the recognition accuracy of the system.

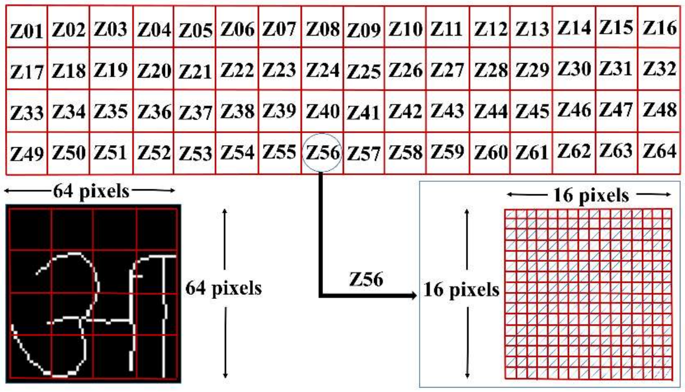

Figure 7.2: Diagonal based features of a Devanagari handwritten word

The following are the steps used for extracting above mentioned features:

Step 1: Input the scanned image of handwritten Devanagari word and convert it into corresponding binary image.

Step 2: After preprocessing, divide it into n number of zones (here $n = 64$ with a pixel size of each zone 16×16) in hierarchical order.

Step 3: Each zone has m number of diagonals (here, $m = 31$). Calculate number of foreground pixels present along with each diagonal, so that sub-feature values of each zone can be extracted.

Step 4: These m sub-features values are averaged to form a single value so that same can be placed in the corresponding zone as its feature. If there is no foreground pixel in the diagonal, then the value of sub-feature of that particular zone is considered as zero.

Step 5: Finally, calculate average featured value by summing up all the above obtained sub-feature values of each diagonal.

Step 6: Sum up all the sub-feature values of each diagonal and calculate average feature values of all sub-values to form a single feature value for the particular zone.

For this work, 85 features of a handwritten word image are extracted by considering above steps.

7.2.3 Centroid-based Features

In centroid based feature extraction method, preprocessed scanned image of handwritten word is divided into n number of zones in similar manner as the case of zoning method. After it, centroid of foreground pixels of each zone are calculated (Kaur et al., 2020). Thereafter, zone-wise distances of each pixel from the centroid are computed. After that, sub-feature value is obtained by average of all the calculated distances. Coordinates of the each centroid may also be considered as one of the vectors in feature set. If a zone has not any foreground pixel, then zero is assigned as feature vector value of corresponding zone. Centroid based features of a Devanagari handwritten word has been depicted in the Fig. 7.3.

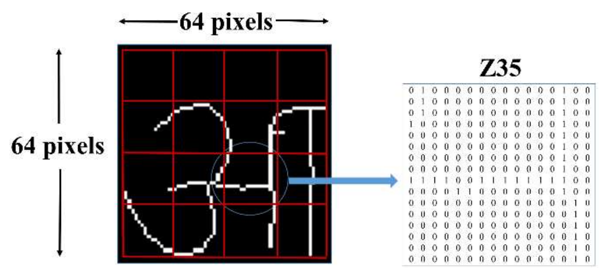

Figure 7.3: Centroid based features of a Devanagari handwritten word

Steps for the same has been outlined below:

Step 1: Input the digitized image of a handwritten word and obtain its binary image.

Step 2: Then, divide the bitmapped word image into n number of equal sized zones.

Step 3: Calculate the centroid of each zone.

Step 4: Obtain the mean distance of each pixels from the centroid in a zone and save it to form a sub-feature for that zone.

Step 6: Obtain a feature vector by repeating above steps for every zone of an image.

Step 7: Feature value can be taken as zero for those zones which does not contain any black pixels.

For this work, 85 features as feature vector of a word image are extracted by applying above mentioned steps.

7.3 GRADIENT BOOSTED DECISION TREE (GBDT) APPROACH

Gradient Boosted Decision Tree (GBDT) is a powerful machine learning method that can effectively combine weak classifiers to create a strong ensemble (Neelakandan and Paulraj, 2020; Wahid et al., 2021). In GBDT, the weak learners are typically individual decision trees. This method optimizes the predictive value of the model through sequential steps during training. Each iteration of the decision tree adjusts the weights assigned to input variables to predict the target value, minimizing the loss function. The trees are connected in a sequential manner, with each tree aiming to reduce the error of the previous tree, as illustrated in Fig. 7.4. This iterative process simplifies the objective and helps obtain an optimal solution. By aggregating the predictions from each stage, GBDT produces a robust learner for handwritten word recognition. This technique significantly enhances the training and learning process, resulting in improved accuracy and performance.

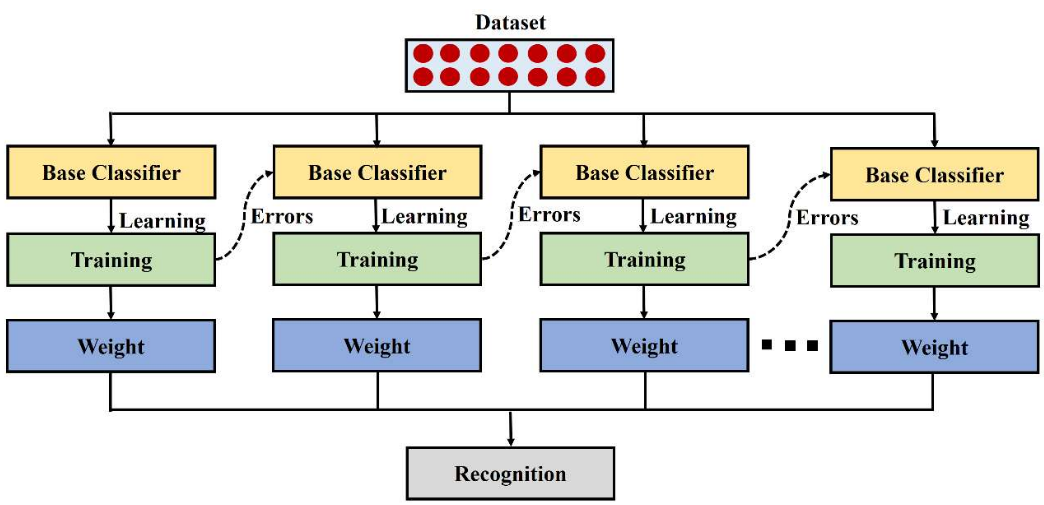

Figure 7.4: Work flow of Gradient Boosted Decision Tree (GBDT)

7.4 EXPERIMENTAL RESULTS AND DISCUSSION

In the section, experiment results based on some statistical features and classification methods considered, have been presented in terms of various performance metrics namely Recognition Accuracy (RA), False Acceptance Rate (FAR), False Rejection

Rate (FRR), F1-Score (FS), Matthew's Correlation Coefficient (MCC) and Area Under the Curve (AUC).

7.4.1 System Performance based on Recognition Accuracy (%)

Experimental results in terms of Recognition Accuracy (RA) for various feature extraction and classification techniques considered for this experimental work, have been depicted in the Table 7.1.

Table 7.1: System performance based on recognition accuracy (%)

Feature Extraction Technique	Classification Techniques			
	K-Nearest Neighbors (KNN)	Decision Tree (DT)	Random Forest (RF)	Gradient Boosted Decision Tree (GBDT)
Uniform Zoning Features (FD1)	70.86%	75.33%	79.68%	82.40%
Diagonal Features (FD2)	71.91%	76.50%	81.01%	83.25%
Centroid Features (FD3)	73.85%	78.00%	81.41%	84.03%
FD1 + FD2	84.50%	87.41%	89.60%	92.16%
FD1 + FD3	85.41%	87.90%	90.03%	92.98%
FD2 + FD3	86.10%	88.58%	90.85%	93.95%
FD1 + FD2 + FD3	86.38%	89.18%	91.60%	94.53%

The following figure (Refer Fig. 7.5) gives graphical representation for the same.

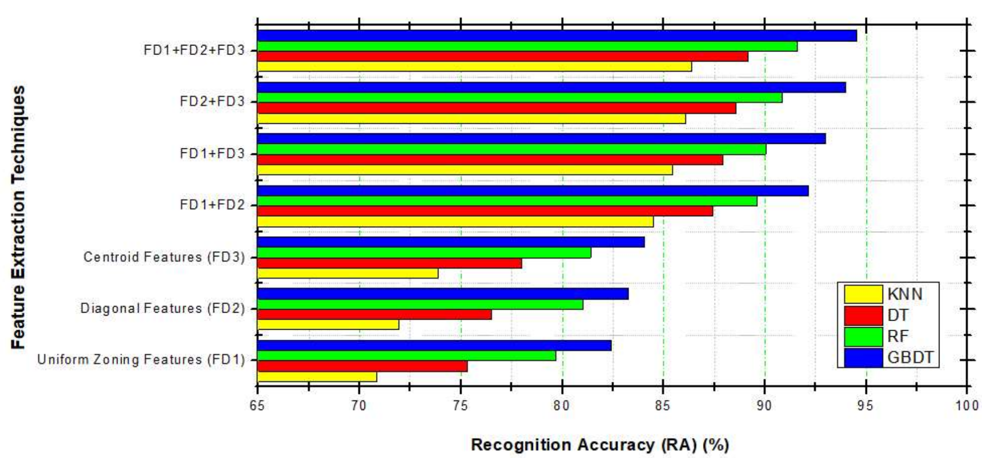

Figure 7.5: Recognition Accuracy (RA) based on various feature extraction and classification methods considered

It has observed that using GBDT classification along with combination of various feature extraction techniques namely uniform zoning, diagonal and centroid based features (FD1+FD2+FD3), maximum recognition accuracy (RA) of 94.53% has been obtained. It can also be seen from table that there is the improvement in recognition accuracy when combination of various statistical features have been considered.

Minimum recognition accuracy of 70.86% is obtained by considering zoning feature extraction and KNN classification techniques. Similar results have been obtained by (Kaur and Kumar, 2021a, 2021b) in their work for handwritten recognition of Gurmukhi words.

7.4.2 System Performance based on FAR (%)

The maximum FAR achieved is 0.59% based on uniform zoning features (FD1) and KNN classification, whereas minimum FAR obtained is 0.11% from combination of all three features and GBDT classification techniques as depicted in Table 7.2. Low value of FAR is desirable for handwritten word recognition applications. Low values of FAR give rise to higher value of FRR. False Acceptance Rate (FAR) results are also presented graphically in Fig. 7.6 for the framework taken for this work.

Table 7.2: System performance based on FAR (%)

Feature Extraction Technique	**Classification Techniques**			
	K-Nearest Neighbors (KNN)	**Decision Tree (DT)**	**Random Forest (RF)**	**Gradient Boosted Decision Tree (GBDT)**
Uniform Zoning Features (FD1)	0.59%	0.50%	0.41%	0.35%
Diagonal Features (FD2)	0.57%	0.47%	0.38%	0.34%
Centroid Features (FD3)	0.53%	0.44%	0.37%	0.32%
FD1 + FD2	0.31%	0.25%	0.21%	0.15%
FD1 + FD3	0.29%	0.24%	0.20%	0.14%
FD2 + FD3	0.28%	0.23%	0.18%	0.12%
FD1 + FD2 + FD3	0.27%	0.22%	0.17%	0.11%

It is observed that three features together (FD1+FD2 + FD3) resulted in the lowest FAR for all classification techniques. The findings highlight the importance of feature selection and classifier choice in achieving lower FAR for the system.

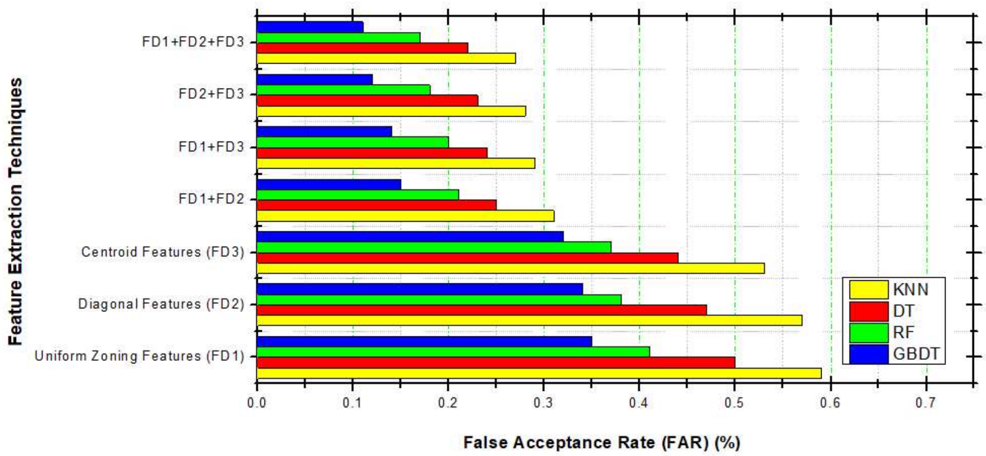

Figure 7.6: False Acceptance Rate (FAR) based on various feature extraction and classification methods considered

7.4.3 System Performance based on FRR (%)

GBDT classifier, scored minimum FRR of 5.46% by considering the combination of all features (FD1+FD2+FD3) for HWR system developed for Devanagari script as given in Table 7.3. FAR and FRR values depend upon each other. As FAR decreases, FRR increases and vice-versa. If the values of both metrics (FAR and FRR) are same, then there exist a point where lines intersect to each other called Equal Error Rate (ERR). False Rejection Rate (FRR) results are also presented graphically in Fig. 7.7 for the framework taken for this work.

Table 7.3: System performance based on FRR (%)

Feature Extraction Technique	Classification Techniques			
	K-Nearest Neighbors (KNN)	Decision Tree (DT)	Random Forest (RF)	Gradient Boosted Decision Tree (GBDT)
Uniform Zoning Features (FD1)	29.13%	24.66%	20.31%	17.60%
Diagonal Features (FD2)	28.08%	23.50%	18.98%	16.74%
Centroid Features (FD3)	26.15%	22.00%	18.58%	15.96%
FD1 + FD2	15.50%	12.58%	10.40%	7.83%
FD1 + FD3	14.58%	12.10%	9.96%	7.01%
FD2 + FD3	13.90%	11.41%	9.15%	6.05%
FD1 + FD2 + FD3	13.61%	10.81%	8.40%	5.46%

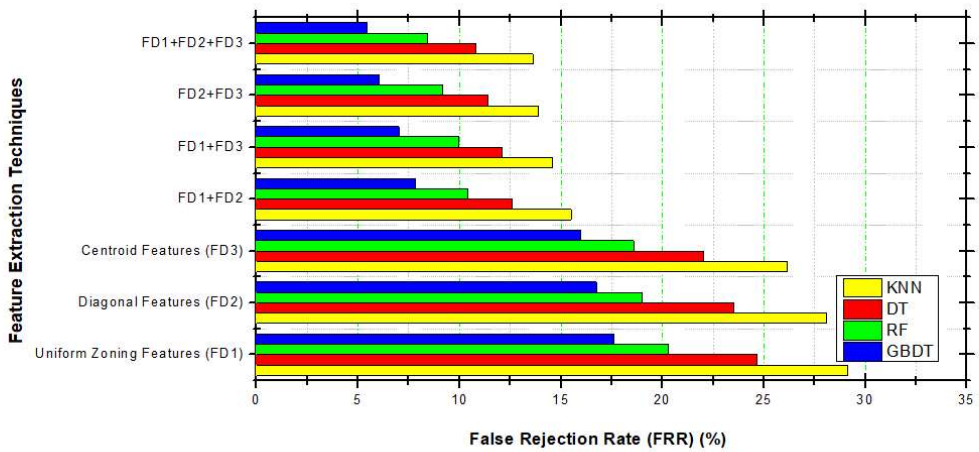

Figure 7.7: False Rejection Rate (FRR) based on various feature extraction and classification methods considered

7.4.4 System Performance based on F1-Score (%)

In Table 7.4, comparative analysis of various feature extraction and classification techniques explored for HWR of Devanagari script is represented in terms of F1-Score (FS). These results are also presented graphically in Fig. 7.8, which clearly indicates the performance of the proposed framework. It has observed that (FD1+FD2+FD3) features and GBDT classification is the better combination as maximum F1-Score of 94.56% is obtained. Because, F1-Score (FS) is the mean of precision and recall, therefore it gives the same amount of weight for both precision and recall. High FS indicates that the system has high precision and recall values.

Table 7.4: System performance based on F1-Score (%)

Feature Extraction Technique	**Classification Techniques**			
	K-Nearest Neighbors (KNN)	**Decision Tree (DT)**	**Random Forest (RF)**	**Gradient Boosted Decision Tree (GBDT)**
Uniform Zoning Features (FD1)	71.19%	75.56%	79.76%	83.32%
Diagonal Features (FD2)	72.20%	76.68%	81.02%	83.21%
Centroid Features (FD3)	74.08%	78.14%	81.39%	84.00%
FD1 + FD2	84.46%	87.43%	89.63%	92.22%
FD1 + FD3	85.42%	87.93%	90.05%	93.05%
FD2 + FD3	86.08%	88.61%	90.88%	94.00%
FD1 + FD2 + FD3	86.36%	89.22%	91.66%	94.56%

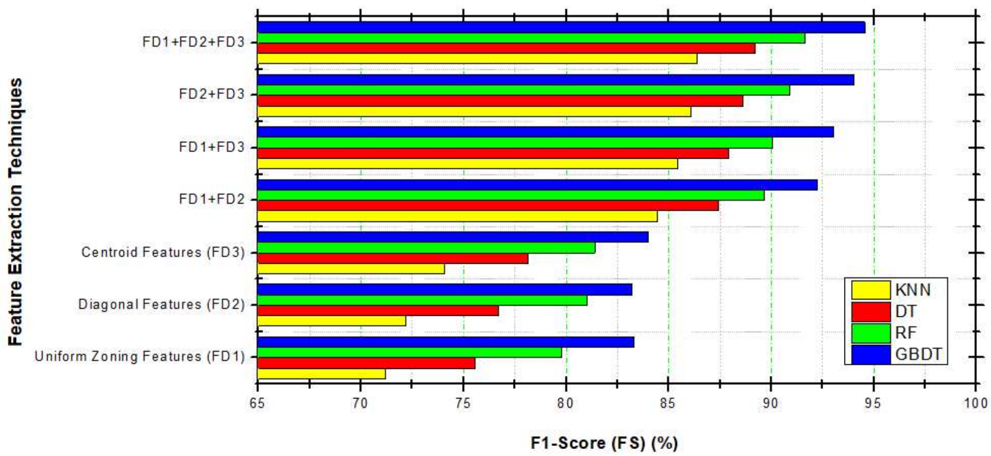

Figure 7.8: F1-Score (FS) based on various feature extraction and classification methods considered

7.4.5 System Performance based on MCC

Maximum achieved MCC value is 0.945 (see Table 7.5) for combination of three features namely uniform zoning, diagonal and centroid based features using GBDT classifier. Whereas, minimum attained MCC values is 0.709 by considering uniform features only along with KNN classifier.

Table 7.5: System performance based on MCC

Feature Extraction Technique	Classification Techniques			
	K-Nearest Neighbors (KNN)	Decision Tree (DT)	Random Forest (RF)	Gradient Boosted Decision Tree (GBDT)
Uniform Zoning Features (FD1)	0.709	0.754	0.798	0.825
Diagonal Features (FD2)	0.720	0.766	0.811	0.833
Centroid Features (FD3)	0.739	0.781	0.815	0.841
FD1 + FD2	0.845	0.875	0.896	0.921
FD1 + FD3	0.855	0.879	0.900	0.929
FD2 + FD3	0.861	0.886	0.908	0.939
FD1 + FD2 + FD3	0.864	0.892	0.916	0.945

The graphical representation of the results is displayed in Fig. 7.9, showcasing the Matthew's Correlation Coefficient (MCC) for different feature extraction and classification techniques. The plot illustrates the performance of each method and its respective MCC value. The graph provides a visual comparison of how various combinations of feature extraction and classification approaches perform in terms of their MCC scores.

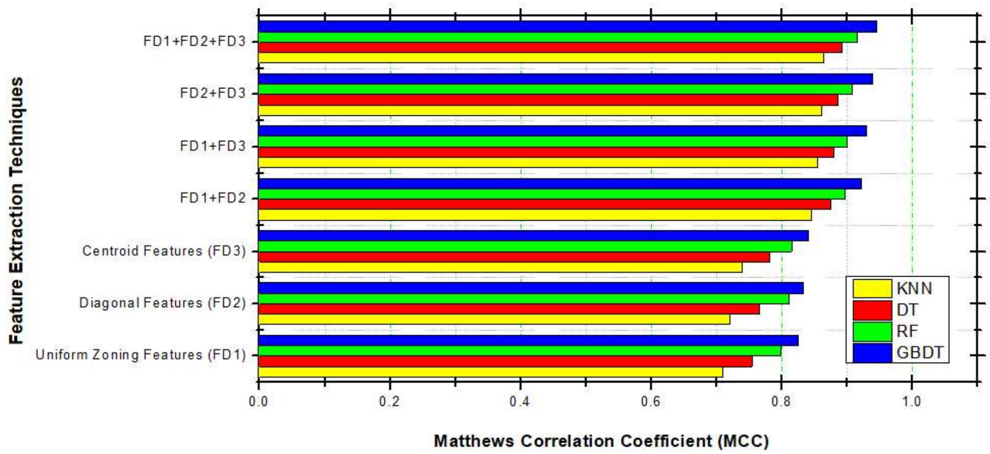

Figure 7.9: Matthew's Correlation Coefficient (MCC) based on various feature extraction and classification methods considered

7.4.6 System Performance based on AUC (%)

Similarly, maximum achieved AUC value is 97.21% (refer Table 7.6) for combination of three features namely uniform zoning, diagonal and centroid based features using GBDT classifier. Higher values of AUC indicates the better performance of the system.

Table 7.6: System Performance based on AUC (%)

Feature Extraction Technique	Classification Techniques			
	K-Nearest Neighbors (KNN)	Decision Tree (DT)	Random Forest (RF)	Gradient Boosted Decision Tree (GBDT)
Uniform Zoning Features (FD1)	85.13%	87.41%	89.63%	91.02%
Diagonal Features (FD2)	85.67%	88.01%	90.31%	91.45%
Centroid Features (FD3)	86.65%	88.77%	90.51%	91.85%
FD1 + FD2	92.09%	93.57%	94.69%	96.00%
FD1 + FD3	92.55%	93.82%	94.91%	96.42%
FD2 + FD3	92.90%	94.17%	95.33%	96.91%
FD1 + FD2 + FD3	93.05%	94.48%	95.71%	97.21%

In the Fig. 7.10, graphical representation of Area Under the Curve (AUC) is depicted based on various feature extraction and classification methods considered for handwritten Devanagari word recognition.

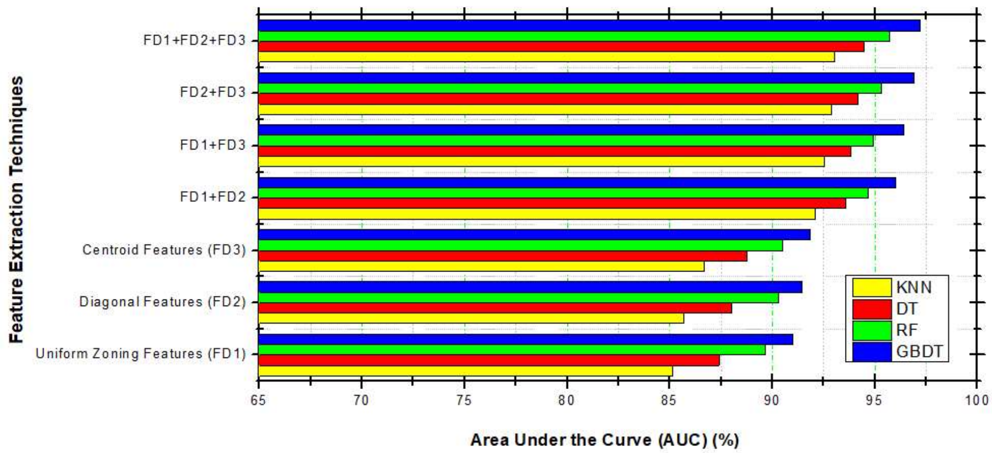

Figure 7.10: Area Under the Curve (AUC) based on various feature extraction and classification methods considered

Experimental study shows that by considering combination of three features namely uniform zoning, diagonal and centroid based features (FD1+FD2+FD3) along with GBDT classifier improve the recognition accuracy and other performance metrics for handwritten Devanagari word recognition system.

7.5 COMPARISON WITH THE STATE-OF-THE-ART WORK AND SYNTACTIC ANALYSIS

In this section, syntactic analysis as a comparison among exiting state-of-the-art work and proposed work is carried out. The proposed system has achieved recognition accuracy of 94.53%, F1-Score of 94.56%, FAR of 0.11%, FRR of 5.46%, MCC of 0.945 and AUC of 97.21%. It can be observed from the Table 7.7 that, present proposed system obtained better and competent performance as compared with other similar systems. It is also to mention that due to the lack of standard database, researchers had performed their experimentation on collected corpus of handwritten words. Hence, it is very challenging and difficult to directly compare the recognition results. Parui and Shaw, (2007) achieved recognition accuracy of 87.71% on a corpus of 10,000 handwritten words of Devanagari script. In (Shaw et al., 2008a, 2008b), authors obtained recognition accuracies of 80.2% and 84.31% using directional chain code and stroke based features, respectively by considering HMM classifier for each and a corpus of 39,700 handwritten words (Devanagari). Using a corpus of 13,000 handwritten Devanagari words, Shaw and Parui, (2010) attained 91.25% of recognition accuracy. They explored stroke and wavelet based features with HMM classification. Singh et al., (2011) in their work achieved maximum recognition accuracy of 93.21% based on Curvelet Transform feature set and KNN classification on a corpus of 28,500 handwritten words (Devanagari). Ramachandrula et al., (2012) obtained maximum recognition accuracy of 91.23% by considering directional element based features and dynamic programming on a database of 39,600 Hindi words.

Furthermore in (Shaw et al., 2014; Shaw et al., 2015), authors obtained recognition accuracies of 79.01% (skeleton and contour based feature set) and 88.75% (DDD and GSC feature set) by considering SVM and multi-class SVM classification methods 39,700 handwritten words (Devanagari). Kumar, (2016) got recognition accuracy of 80.8% (neighbor pixel weight and gradient feature set) on a corpus of more than 3,500 Devanagari words using MLP classifier. Further, based on MLP classification, Malakar et al., (2017) explored low-level features and obtained recognition accuracy of 96.82% (corpus of 4,620 Hindi words). Bhunia et al., (2018) achieved recognition accuracy of above 60% using PHOG (Pyramid Histogram of Oriented Gradients) feature set and HMM and SVM classification on a database of 3,856 (Bangla); 3,589 (Devanagari) and 3,142 (Gurumukhi) words. On corpus of 7,500 handwritten Bangla words, Ghosh et al., (2019) obtained 93% of recognition accuracy by exploring gradient and modified SCF features and MA-based wrapper filter selection approach along with MLP classification. Whereas, on a database of 12,000 handwritten Bangla words, Malakar et al., (2020a) obtained higher recognition accuracy of 95.30% based on gradient-based & elliptical feature set and MLP classifier.

Moreover, Kaur and Kumar, (2021a) obtained recognition accuracy of 91.66% (on a database of 40,000 handwritten Gurumukhi words) by exploring zoning features and XGBoost. It is evident from experimentation and syntactic analysis that proposed work achieved better recognition results by incorporating gradient boosted decision tree with the combination of statistical features for recognition of handwritten word recognition system for Devanagari words. In Fig. 7.11, the confusion matrix is depicted when a combination of zoning, diagonal and centroid based features along with GBDT classifier, have considered.

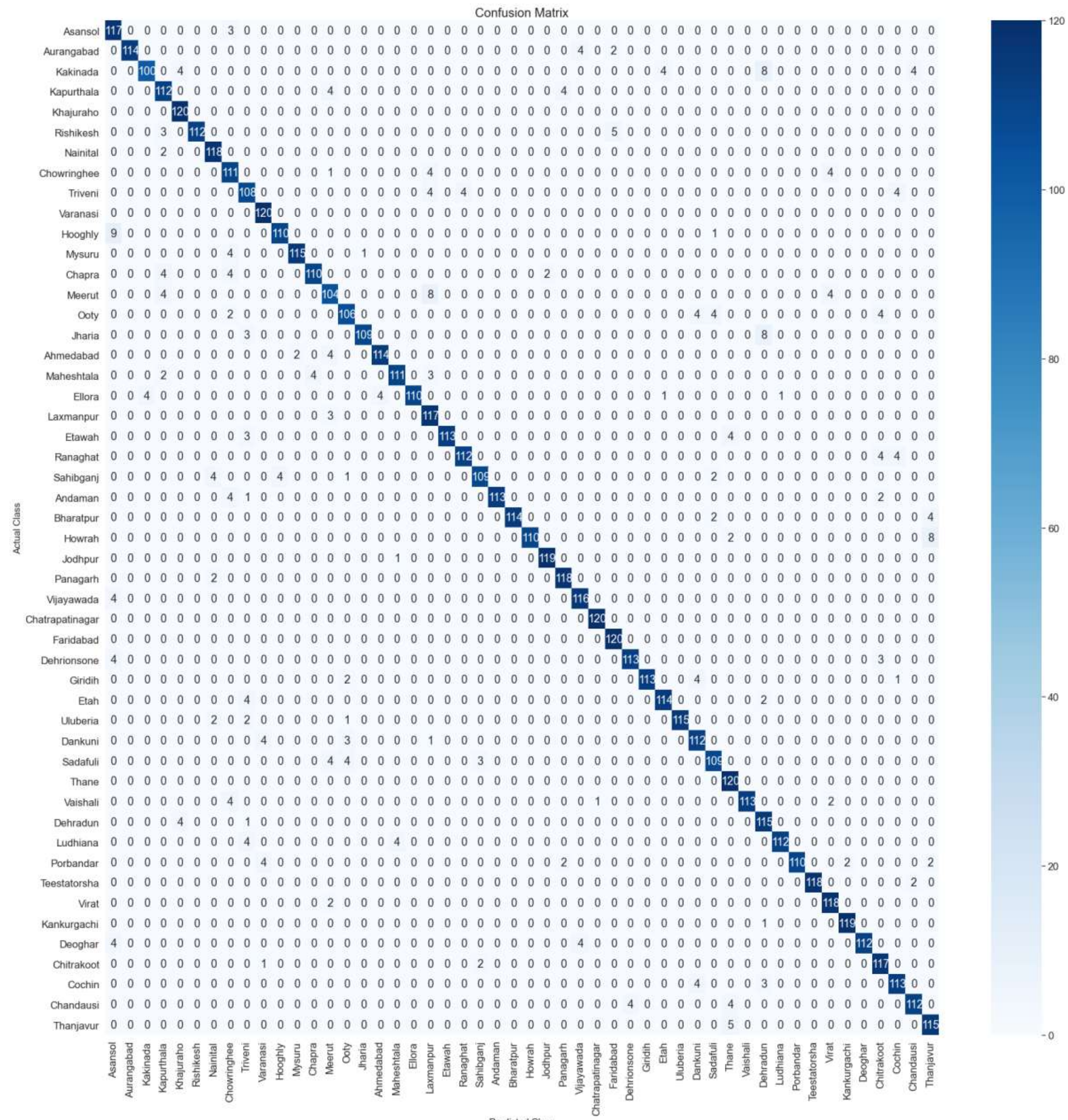

Figure 7.11: Visualization of confusion matrix

It shows correlation amongst predicted class (word) and actual class (word). It is a useful tool for analyzing the effectiveness of a classification phase of handwritten word recognition system. A confusion matrix is an $N \times N$ matrix, where N denotes target classes i.e. predicted and actual classes (for this work, $N = 50$). It is usually used to visualize and summarize the performance of a classification algorithm i.e. how well a classification algorithm performs. Every entry in a confusion matrix indicates whether the proposed system recognize the word-classes correctly or incorrectly. As a result, the confusion matrix makes it evident how the system is doing while categorizing or recognizing various word-classes. Thus, confusion matrix is significant tool as it provides comprehensive understanding (it compares the actual word-classes with predicted word-classes) of how well the proposed system is performing. The confusion matrix of size 50×50 (refer Fig. 7.11), summarizes the experimental results for handwritten Devanagari word recognition system (for each of 50 word-classes considered for this work). Each row represents the frequency count of handwritten words recognized or identified by the system with darker colors associated with higher frequency counts. The handwritten Devanagari words can either be recognized correctly (predicted word-class is same as actual word-class) or incorrectly (predicted word-class is not same as actual word-class). In this work, maximum recognition accuracy of 94.53% has been computed from the confusion matrix.

Table 7.7: Comparison of proposed work with existing methodologies

Author(s)	**Script or Language**	**Dataset (Words)**	**Approach**		**Recognition Accuracy (%)**
			Feature Extraction	**Classification**	
Parui and Shaw, (2007)	Devanagri	10,000	Stroke-based	HMM	87.71%
Shaw et al., (2008a)	Devanagri	39,700	Directional Chain Code-based	HMM	80.20%
Shaw et al., (2008b)	Devanagri	39,700	Stroke-based	HMM	84.31%
Shaw and Parui, (2010)	Devanagri	13,000	Stroke based *(Stage-1);* Wavelet-based *(Stage-2)*	HMM *(Stage-1);* Modified Byes *(Stage-2)*	91.25% *(Stage-2)*
Singh et al., (2011)	Devanagri	28,500	Curvelet Transform-based	SVM and KNN	85.60% *(SVM);* 93.21% *(KNN)*
Ramachandrula et al., (2012)	Hindi	39,600	Directional Element-based	Dynamic Programming	79.94% *(30 Vocabulary Words);* 91.23% *(10 Vacbulary Words)*
Shaw et al., (2014)	Devanagri	39,700	Combination of Skeleton and Contour-based	SVM	79.01%

Shaw et al., (2015)	Devanagri	39,700	DDD and GSC-based	Multiclass SVM	88.75%
Kumar, (2016)	Devanagari	More than 3,500	Chain Codes, Cumulative Histograms, Gradient, Neighbor Pixel Weight-based	MLP	80.80% *(for Two Character Words)* 72.00% *(for Six Character Words)*
Malakar et al., (2017)	Hindi	4,620	Low-level features	MLP	96.82%
Bhunia et al., (2018)	Bangla, Devanagari and Gurumukhi	3,856; 3,589 and 3,142	PHOG-based	HMM *(for Middle-Zone)* SVM *(for Upper/Lower Zone)*	Above 60.00%
Ghosh et al., (2019)	Bangla	7,500	Gradient and Modified SCF; MA-based Wrapper Filter Selection Approach	MLP	93.00%
Malakar et al., (2020a)	Bangla	12,000	Gradient and Elliptical-based	MLP	95.30%
Kaur and Kumar, (2021a)	Gurumukhi	40,000	Zoning-based	XGBoost	91.66%
Proposed Study	Devanagari	20,000	Combination of Uniform Zoning, Diagonal and Centroid-based	Gradient Boosted Decsion Tree	94.53%

7.6 CHAPTER SUMMARY

Recognition of Indic and Non-Indic handwritten words has been an active and popular research area from past few years due to its variety of real-time potential applications including postal automation etc. The aim of this chapter is to present a performance based analytical study of different features and classifiers for the recognition of handwritten Devanagari words. In this chapter, unıform zoning, diagonal and centroid based features are explored along with various classifiers namely KNN, decision tree, random forest and gradient boosted decision tree for experimental work. Experimentation is also performed by considering combination of above mentioned features and classifiers. Based on the experimental study, maximum recognition accuracy of 94.53%, F1-Score of 94.56%, FAR of 0.11%, FRR of 5.46%, MCC of 0.945 and AUC of 97.21% are achieved for the handwritten Devanagari word recognition. Moreover, comparison with existing methods and syntactic analysis are presented for the assessment of recognition performance in terms of the recognition accuracy.

Overall, proposed system performed good and competent as compared with other existing similar state-of-the-art systems for recognizing the handwritten Devanagari words.