"Research is to see what everybody else has seen and think what nobody has thought"
-Albert Szent-Gyorgyi

Chapter 8

VGG16: AN EFFICIENT APPROACH FOR OFFLINE HANDWRITTEN DEVANAGARI WORD RECOGNITION USING DEEP FEATURES AND XGBOOST

8.1 INTRODUCTION

The proposed method describes a holistic approach to recognizing offline handwritten Devanagari words which belongs to fifty different classes. The recognition steps include: digitization, preprocessing, feature extraction approach (VGG16) and classification approaches (Gaussian NB, XGBoost and RF). In the proposed approach, firstly digital image of the offline handwritten document is generated by scanning. Then, handwritten words are obtained from scanned documents using suitable preprocessing operations. After that, features are extracted from word images using VGG16 (deep features) as feature extractor considering holistic based approach to generate desired feature database. Thereafter, words are classified and recognized using different classifiers namely Gaussian NB, XGBoost and Random Forest. Finally, in order to test the performance of the proposed system, various performance evaluators namely Recognition Accuracy (RA), Precision (PR), Recall (RL), F1-Score (FS) and Area Under Curve (AUC) are computed. To best of present knowledge, above mentioned techniques have been considered for the very first time to recognize handwritten words written in Devanagari script in this work. In this entire process, firstly the system is trained using the training dataset and then its performance has been analyzed using the testing dataset followed by the above steps.

In Section 8.2, VGG16 as feature extractor has been discussed. The description about XGBoost (Extreme Gradient Boosting) approach is given in Section 8.3. Whereas, experimental results and discussion is presented in Section 8.4. Comparison with the state-of-the-art work and syntactic analysis is outlined in Section 8.5. Finally, chapter summary is given in Section 8.6.

8.2 VGG16 AS FEATURE EXTRACTOR

The architecture of VGG16 (Visual Geometry Group) consists of convolutional (1-13), MaxPooling (A-E) and fully connected (14-15) layers followed by a single softmax (16) layer for the output (Simonyan and Zisserman, 2014). As 16 layers have weights hence it is named VGG16. Out of five blocks, the first two blocks (Block 1-2) have two convolutional layers while the other three blocks (Block 3-5) have three convolutional layers, each followed by the MaxPooling layer as presented in Fig. 8.1. The feature extraction part of the model is considered from the input layer to the last max-pooling layer while the remaining part of the network is considered as a classification part of the model. In this work, VGG16 is used as a feature extractor only so that further recognition of handwritten Devanagari words be carried out using different classifiers.

Hence, the last dense layers of the VGG16 model have been removed so that it can be used as a feature extractor (without classifier) only.

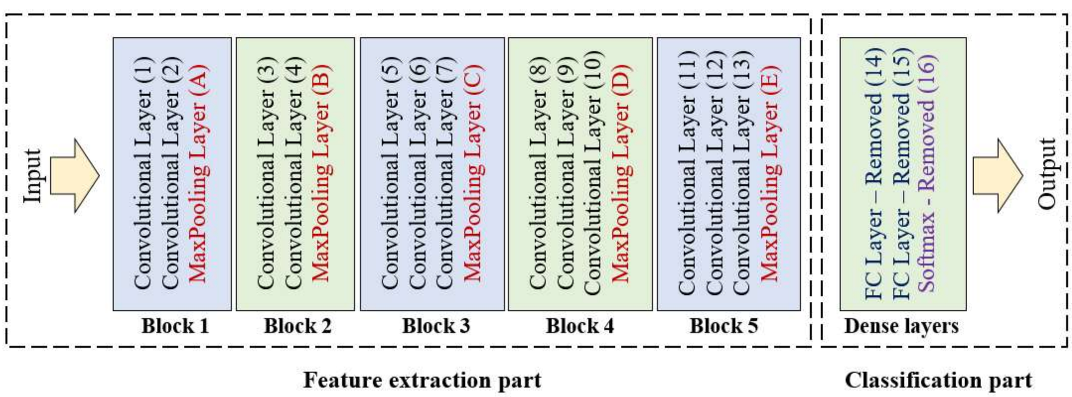

Figure 8.1: VGG16 model without dense layers

After removing the top dense layers (by setting parameter include_top=false), extraction of features shall be carried out using the last MaxPooling layer (Block 5) of VGG16. The convolutional layer consists of the number of channels starting from 64 in the first layer. After every MaxPooling layer, the number is increased by a factor of 2 until it reaches 512. The last two Fully Connected (FC) layers and softmax layers have been removed (classification part), so that VGG-16 shall return dimensional feature representative vectors only (Islam et al., 2019b). After extracting these features, a feature vector has been prepared so that these can be used for word classification/recognition.

8.3 XGBOOST (EXTREME GRADIENT BOOSTING) APPROACH

XGBoost (eXtreme Gradient Boosting) is a supervised machine learning approach that can be used for handwritten word or text classification. It is basically a gradient boosted decision tree implementation that may be considered to achieve good speed and performance of the model. In gradient boosting, to predict the classification errors new models are created so that same can be added with existing one to make the final prediction. Thus, it results a final model depending on the combination of individual models. As it uses a gradient descent algorithm, to minimize the loss during addition of new models hence named as gradient boosting (Ren et al., Li, 2017).

This classifier combines several classifications and regression trees. Consider a database containing ‘p’ samples and ‘q’ features, $D = \{(x_i, y_i\}$ ($|D| = x_i \in S^{p\times q}, y_i \in S^p$), the mathematically ensemble algorithm may expressed using Eq. 8.1 as follows:

$$\hat{y}_i = \sum_{n=1}^{n} f_n(x_i), f_n \in S \tag{8.1}$$

Where $'n'$ represents the number of trees, $'f'$ denotes function in functional space $'S'$, where $'S'$ indicates the set of comprising all possible classification/regression trees. The training time depends upon the number of dataset-classes because the generation of trees depends upon the number of labels/categories.

8.4 EXPERIMENTAL RESULTS AND DISCUSSION

To analyze the performance of the proposed system, experimental results are presented in terms of various performance evaluators considering a corpus/dataset of 15,000 handwritten Devanagari words. For this work, handwritten Devanagari word datasets is divided into three strategic schemes namely X, Y, and Z; as depicted in the following Table 8.1.

Table 8.1: Dataset partitioning schemes

Strategic-schemes	Partitioning	Training dataset (words)	Testing dataset (words)
X	90:10	13,500	1,500
Y	80:20	12,000	3,000
Z	70:30	10,500	4,500

In X strategic scheme, training datasets (words) are considered as 90% of the dataset whereas 10% of datasets are considered as testing datasets (words). For Y strategic scheme, 80% of the dataset is taken as the training dataset (words) and 20% as a testing dataset (words). In the Z strategic scheme, 70% of the dataset is considered as a training dataset (words) and 30% as a testing dataset (words). Experiments are performed using VGG16 (deep features) as feature extractor and three classification approaches namely Gaussian naive based (Gaussian NB), XGBoost and Random Forest (RF) for recognition of handwritten word images. In the following sub-section, system performance analysis in terms of various performance evaluators is presented.

8.4.1 Performance Analysis based on Recognition Accuracy (%)

Table 8.2 presents the performance analysis based on recognition accuracy (%) for different strategic schemes and classification techniques. The strategic schemes, represented by X, Y and Z, indicate different train-test data splitting ratios, with X (90:10) indicating 90% training data and 10% testing data, Y (80:20) indicating 80% training data and 20% testing data, and Z (70:30) indicating 70% training data and 30% testing data.

Table 8.2: Performance analysis based on recognition accuracy (%)

Strategic-Schemes	Classification Techniques		
	Gaussian Naive Based (Gaussian NB)	eXtreme Gradient Boosting (XGBoost)	Random Forest (RF)
X (90:10)	82.70%	93.10%	95.00%
Y (80:20)	82.06%	93.00%	94.80%
Z (70:30)	80.40%	92.80%	94.40%

Table 8.2 indicates that the highest accuracy of 95.00% is achieved for X strategic-scheme and while the second-highest accuracy of 94.80% is achieved for Y strategic-scheme using RF classifier.

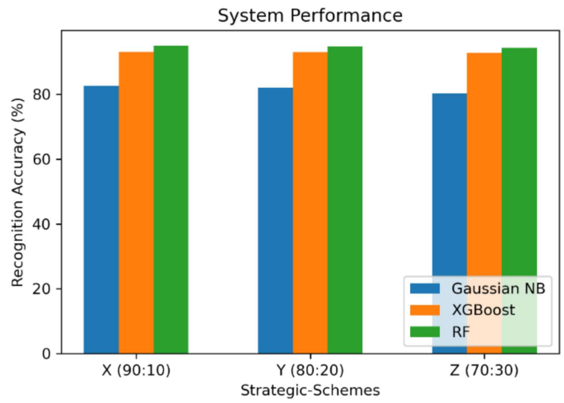

Figure 8.2: System performance in terms of recognition accuracy (%)

Based on the graphical results as presented in Fig. 8.2, it can be observed that the Random Forest classification technique consistently performs the best among the three techniques across all strategic schemes. XGBoost also demonstrates high accuracy, while Gaussian NB generally achieves slightly lower accuracy compared to the other techniques.

8.4.2 Performance Analysis based on Precision (%)

The performance analysis presented in Table 8.3 provides valuable insights into the precision performance of different classification techniques and strategic-schemes. These findings shall guide the selection of the most suitable approach for a given recognition task, considering the trade-off between precision and other evaluation metrics.

Table 8.3: Performance analysis based on precision (%)

Strategic-Schemes	Classification Techniques		
	Gaussian Naive Based (Gaussian NB)	eXtreme Gradient Boosting (XGBoost)	Random Forest (RF)
X (90:10)	85.01%	93.40%	95.18%
Y (80:20)	84.09%	93.29%	95.23%
Z (70:30)	83.47%	93.32%	94.59%

From Table 8.3, it can be summarized that the highest precision of 95.23% is achieved for the Y strategic-scheme and the second-highest precision of 95.18% is achieved for X strategic-scheme using RF classifier. The graphical representation of the system performance in terms of Precision (%) is given in Fig. 8.3.

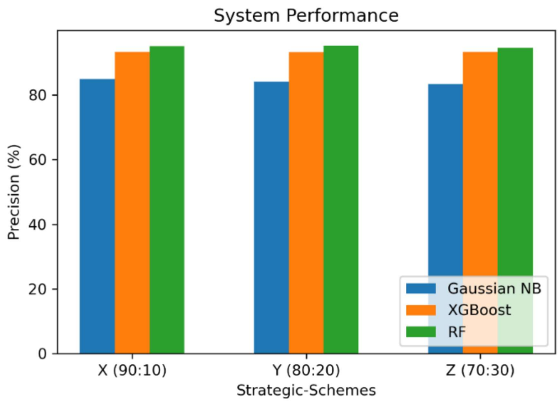

Figure 8.3: System performance in terms of precision (%)

From the results, it can be observed that Random Forest consistently achieves the highest precision across all strategic-schemes. XGBoost also demonstrates high precision, while Gaussian NB generally achieves slightly lower precision compared to the other techniques. A higher precision value indicates a lower number of false positives, which is desirable in many applications. It implies that the systems using Random Forest and XGBoost are better at correctly identifying positive instances, while Gaussian NB may have a slightly higher rate of false positives.

8.4.3 Performance Analysis based on Recall (%)

Table 8.4 presents the performance analysis based on recall (%) for different strategic schemes and classification techniques. The strategic schemes X (90:10), Y (80:20), and Z (70:30) represent different ratios of training and testing data, while the classification techniques include Gaussian Naïve Based (Gaussian NB), eXtreme Gradient Boosting (XGBoost) and Random Forest (RF). The recall metric measures the ability of the system to correctly identify positive instances.

Table 8.4: Performance analysis based on recall (%)

Strategic-Schemes	Classification Techniques		
	Gaussian Naive Based (Gaussian NB)	eXtreme Gradient Boosting (XGBoost)	Random Forest (RF)
X (90:10)	82.70%	93.10%	95.00%
Y (80:20)	82.06%	93.00%	94.80%
Z (70:30)	80.40%	92.80%	94.40%

In terms of recall, the RF classifier achieved the maximum value of recall as 95.00% for X strategic-scheme (Refer Table 8.4), while it is 94.80% for Y strategic-scheme.

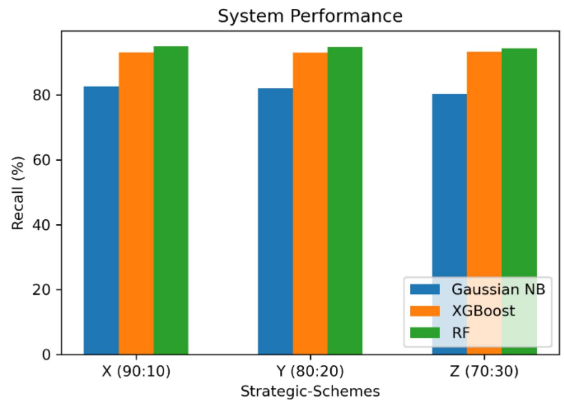

Figure 8.4: System performance in terms of recall (%)

Overall, Random Forest consistently achieves the highest recall values, followed by XGBoost, while Gaussian NB exhibits slightly lower recall rates (refer Fig. 8.4). These findings highlight the strengths of Random Forest and XGBoost in correctly identifying positive instances, providing valuable insights for selecting suitable approaches in recognition tasks.

8.4.4 Performance Analysis based on F1-Score (%)

Table 8.5 presents the performance analysis based on the F1-Score (%) for different strategic-schemes and classification techniques include Gaussian Naive Based (Gaussian NB), eXtreme Gradient Boosting (XGBoost), and Random Forest (RF). The F1-Score is a measure of the system's accuracy, taking into account both precision and recall. Table 8.5 depicts that the RF classifier gained the maximum F1-Score of 94.99% for X strategic-scheme, while for Y strategic-scheme, F1-Score is the second highest of value 94.78%.

Table 8.5: Performance analysis based on F1-Score (%)

Strategic-Schemes	Classification Techniques		
	Gaussian Naive Based (Gaussian NB)	eXtreme Gradient Boosting (XGBoost)	Random Forest (RF)
X (90:10)	82.75%	93.11%	94.99%
Y (80:20)	82.01%	93.08%	94.78%
Z (70:30)	80.13%	92.80%	94.38%

These results are presented graphically in Fig. 8.5, where it is evident that Random Forest consistently achieves the highest F1-Scores among all the strategic schemes. XGBoost also exhibits competitive performance, closely trailing Random Forest. However, Gaussian NB shows slightly lower F1-Scores in comparison to the other classification techniques.

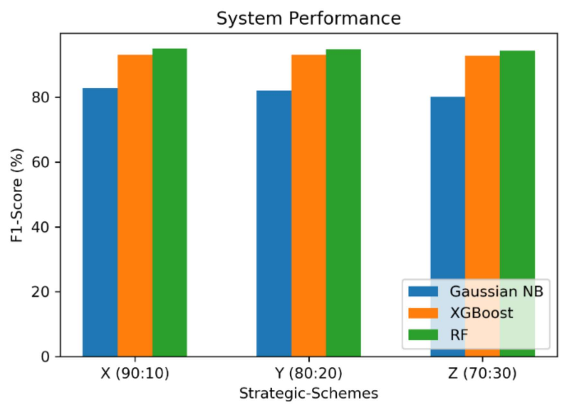

Figure 8.5: System performance in terms of F1-Score (%)

It has been gathered that Random Forest and XGBoost are effective in achieving a balance between precision and recall, resulting in higher F1-Scores. Gaussian NB, although performing relatively lower, still provides reasonable accuracy. This analysis can be used to select the most suitable classification technique based on their specific requirements in handwritten word recognition tasks.

8.4.5 Performance Analysis based on AUC (%)

The performance analysis based on the Area Under the Curve (AUC) metric for different strategic-schemes and classification techniques is presented in Table 8.6. AUC is a widely used evaluation metric in machine learning that measures the overall performance of a system. Table 8.6 gives that the highest AUC of 99.94% is achieved for Z strategic-scheme and while the second highest AUC of 99.94% is achieved for the X strategic-scheme using XGBoost classifier.

From the given table, it can be observed that all three classification techniques, Gaussian NB, XGBoost and Random Forest, achieve high AUC scores across all strategic schemes. XGBoost consistently demonstrates the highest AUC scores, followed closely by Random Forest. Gaussian NB shows slightly lower AUC scores compared to the other techniques as depicted graphically in Fig. 8.6.

Table 8.6: Performance analysis based on AUC (%)

Strategic-Schemes	Classification Techniques		
	Gaussian Naive Based (Gaussian NB)	eXtreme Gradient Boosting (XGBoost)	Random Forest (RF)
X (90:10)	91.17%	99.94%	99.84%
Y (80:20)	90.85%	99.85%	99.87%
Z (70:30)	90.00%	99.94%	99.83%

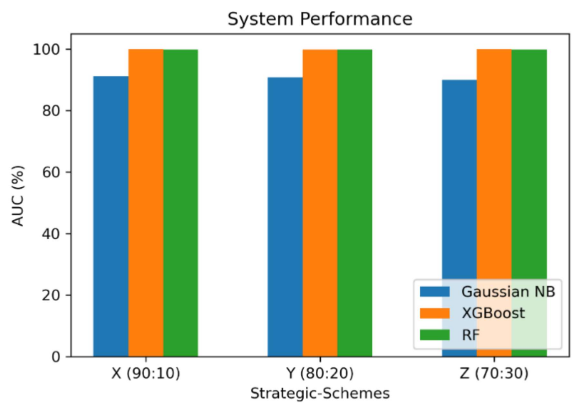

Figure 8.6: System performance in terms of AUC (%)

These results indicate that XGBoost and Random Forest exhibit better discriminative power and are capable of accurately distinguishing between different classes in the dataset. Gaussian NB, although performing slightly lower in terms of AUC, still achieves reasonably good performance in terms of overall classification accuracy. Overall, the high AUC scores obtained by the classification techniques indicate their effectiveness in the task of classification and their ability to produce reliable and robust predictions.

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

In the literature, various existing approaches have been discussed that were used for the recognition of various handwritten scripts/languages like Arabic, Bangla, Devanagari/Hindi and Gurmukhi. Researchers used various feature extraction approaches such as stroke based, wavelet based, curvelet transform based, elliptical features, Directional Distance Distribution (DDD) features, Gradient-Structural-

Concavity (GSC) features, Pyramid Histogram of Oriented Gradients (PHOG) features and classification approaches such as Hidden Markov Model (HMM), Modified Byes, Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Dynamic Programming, Multi-Layer Perceptron (MLP) and Multi-Class SVM. The comparative analysis of these approaches with the proposed approach has been carried out in terms of recognition accuracy (%) and presented in the Table 8.7.

Table 8.7: Comparative analysis of the proposed work with some other existing approaches

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 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)*
Bhowmik, Roushan, et al., (2014)	Bangla	1,020	Elliptical-based	MLP	85.88%
Bhowmik et al., (2015)	Bangla	2,754	Concentric Rectangles and Convex Hull-based	Neural Network based	84.74%
Kadhm and Hassan, (2015)	Arabic Handwriting Database (AHDB)	2,913	Integration of using Multi Scale-based	SVM	96.31%
Shaw et al., (2015)	Devanagri	39,700	DDD and GSC-based	Multiclass SVM	88.75%
Kumar, (2016)	Devanagari	More than 3500	Chain Codes, Cumulative Histograms, Gradient, Neighbor Pixel Weight-based	MLP	80.80% *(for Two Character Words)* 72.00% *(for Six Character Words)*

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	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 Approach	Devanagri	15,000	VGG16	Gaussian NB, XGBoost and RF	82.70% 93.10% 95.00%

- Using stroke based features and HMM classification technique, Parui and Shaw, (2007) achieved accuracy for recognition of Handwritten Devanagari words as 87.71% (Training) and 82.89% (Testing) for a dataset of 7,000 (Training samples) and 3,000 (Testing samples). Shaw and Parui, (2010) improved the accuracy as 85.57% (Testing) and 91.25% (Training) considering stroke based features, wavelet, HMM and modified Byes techniques.
- Whereas, Kumar, (2016) gained 80.80% and 72.00% accuracies for two and six character words respectively on a dataset of more than 3,500 words by using the concept of chain codes, cumulative histograms, Gradient based features and MLP classification.
- On slightly larger sized dataset, Singh et al., (2011) achieved accuracy as 85.60% (SVM) and 93.21% (KNN) considering Curvelet transform and 28,500 words in dataset. While, Shaw et al., (2015) achieved accuracy of 88.75% by employing DDD, GSC and Multi-class SVM on a corpus of 39,700 words. However, Ramachandrula et al., (2012) gained accuracies of 79.94% and 91.23% for a dataset having 30 and 10 sized-vocabulary words respectively.
- For a dataset of 1,020 handwritten Bangla words, Bhowmik et al., (2014) achieved 85.88% accuracy by considering elliptical features. Further, in 2015, by increasing the size of dataset upto 2,754 words, same authors achieved

84.74% accuracy using concentric rectangles, convex hull-based features and NN. Moreover, Ghosh et al., (2019) improved accuracy upto 93.00% using Gradient-based features and modified SCF on the dataset of 7,500 handwritten Bangla words.

- On Arabic handwriting dataset (AHDB) of 2,913 words, Kadhm and Hassan, (2015) achieved 96.31% accuracy by integration multi scale features and SVM. However, Bhunia et al., (2018) worked on dataset of three scripts/languages namely Bangla, Devanagari and Gurumukhi handwritten words of 3856, 3589 and 3142 respectively. By using PHOG feature, HMM and SVM techniques, authors achieved accuracy above 60.00%.
- Malakar et al., (2020) obtained 95.30% accuracy using gradient-based and elliptical features and MLP classifier using corpus of 12,000 handwritten Bangla words. While, Kaur and Kumar, (2021a) obtained 91.66% recognition accuracy using zoning features and XGBoost approach from a database of 40,000 samples of Gurumukhi words.
- Comparatively, proposed approach for recognition of Devanagari handwritten words achieved recognition accuracies of 82.70%, 93.10% and 95.00% using Gaussian NB, XGBoost and RF classifiers respectively. The proposed approach gained a comparable recognition rate using XGBoost and RF classifiers in comparison to the above mentioned approaches developed for the recognition of handwritten words.

8.6 CHAPTER SUMMARY

In this chapter, an efficient offline handwritten Devanagari word recognition system using deep features (VGG16) is presented that results in better recognition accuracy. This is due to the recognition ability of an HWR system depending on the quality of extracted features. A holistic approach treats the word as a single/indivisible entity for feature extraction and recognition purposes. Performance comparison of three classifiers (Gaussian NB, XGBoost and Random Forest) have been done in terms of various performance metrics namely recognition accuracy (%), precision (%), recall (%), F1-Score (%) and area under curve (%) to test their suitability for recognition. This approach finds its numerous real-time applications such as postal automation, signature verification, bank cheque processing and writer identification.

It has been observed that the random forest (RF) based classification technique achieved the maximum accuracy (95.00%), recall (95.00%), precision (95.23%) and F1-Score (94.99%) and while XGBoost based classification scored maximum AUC (99.94%). Experimental results depict that XGBoost classification also performing better than other existing techniques. Moreover, to state the importance of this work, the comparative analysis of the proposed approach with exiting approaches has also been presented.