Bengali Handwritten Characters Recognition

Bornomalar Bondhu
Dipankar Dey(dipankardey02476@gmail.com)
Saikat Kumar Ghosh(saikatghosh2782001@gmail.com)

May 1, 2024

Abstract

This project is going to explore the feasibility of using traditional supervised machine learning techniques for Bengali Handwritten Character Recognition (BHCR). We will use a diverse dataset of handwritten Bengali characters, classical algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests etc. which will be trained and evaluated. Through rigorous performance assessment using standard metrics, including accuracy, precision, the effectiveness of these classical methods will be demonstrated. Feature selection and hyperparameter tuning might be used to further enhance model performance. At the end we will assess and compare the accuracy given by our classical Machine learning techniques with other state of the art techniques .

1 Introduction

The objective of this project is to leverage machine learning techniques to classify handwritten Bengali characters. The Bengali script comprises 11 vowels and 39 consonants, totaling 50 basic characters. By employing machine learning algorithms, we aim to develop a robust classification system capable of accurately identifying and categorizing handwritten Bengali characters into their respective vowels and consonants categories. The complexity of handwritten character recognition, particularly in Bengali, poses significant challenges due to variations in size, shape, and individual writing styles. These challenges are increased by similarities in character shapes, and variations in strokes. Bengali, as one of the world's most spoken languages with rich cultural heritage, demands attention in automatic character recognition. Addressing these challenges through machine learning holds promise for enhancing recognition accuracy and advancing linguistic technology. Here we will be using the classical machine learning models to achieve our objective and will try to get as close as possible with the accuracy given by popular deep learning models by incorporating more complex datasets in our study.

2 Literature review

While starting to work in Bengali handwritten character classification of primary Bengali characters we found that most of the previous work has been focused for classification of Bengali numerals as compared to primary Bengali characters. Though our work was focused on primary character classification, still we did some review of the article titled:"Two Decades of Bengali Handwritten Digit Recognition A Survey" by A. B. M. Ashikur Rahman et al[8] , where they gave an overview of the work that has been done over last two decades for classifying Bengali numerals.

For primary character classification we reviewed the article "A machine learning approach for Bengali handwritten vowel character recognition" by Shahrukh Ahsan et al[1] where they have used SVM to classify Bengali vowels and "BornoNet: Bangla Handwritten Characters Recognition Using Convolutional Neural Network" by Akm Shahariar Azad Rabby et al[5] where they have proposed a CNN based model to classify the primary Bengali characters.

Moreover for dataset preparation and preprocessing we followed "A Universal Way to Collect and Process Handwritten Data For Any Language" by AKM Shahariar Azad Rabby et al[6].

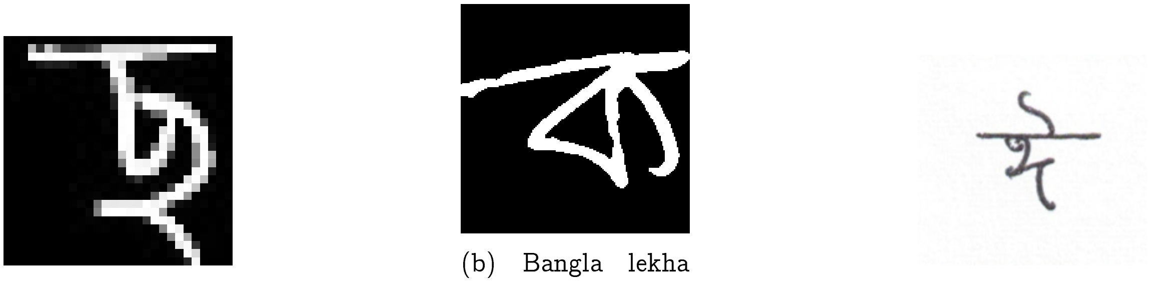
The main difference between our proposed method and the work done by others is that we have used pixel based values as features and deployed classical machine learning techniques to classify the characters to achieve the accuracy as close to what others have achieved.

3 Proposed methodology

The proposed method here is divided into three parts. The first and second part is about the dataset we have used and regarding the preparation of the dataset for training and testing for which we have used Ekush[7] dataset, Bangla lekha isolated dataset and our own collected primary dataset. The third part is the core implementation part where we have used classical ML algorithms to classify the characters.

3.1 Dataset

Here we have used 3 datasets Ekush[7], Bangla lekha isolated[4] and our own primary dataset. Though among them Ekush and Bangla lekha isolated consists of Bangla characters, numerals and compound characters, but for our purpose we have considered only the primary Bengali characters. Ekush consists of 149,341 images, Bangla isolated consists of 86,458 images and our own collected dataset has 2500 images.



(a) Ekush Image

(b) Bangla lekha Image

(c) Our own Image

Figure 1: Datasets

3.2 Dataset Preparation

This section describes the steps for dataset preparation like collecting the handwritten raw images, deriving the pixel values from it and then converting and storing the images in proper numpy format for ease of accessing. Figure-1 gives an overview of the data preparation steps:

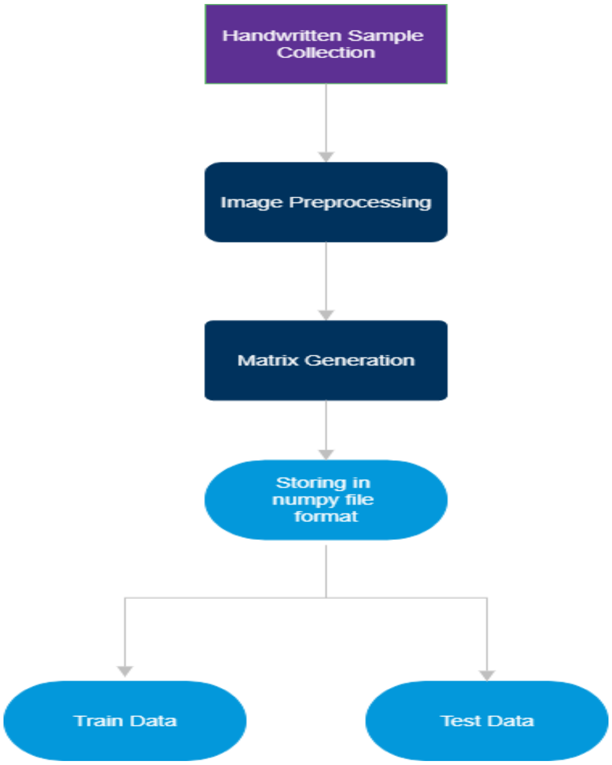


Figure 2: Data Preparation Steps

3.2.1 Data Collection

Through out our work we have used 3 different datasets which are i)Ekush[7], ii)Bangla lekha isolated[4] and iii)Our own collected dataset. For the data collected by us from 50 individuals we had to prepare the forms in a proper format and then extract the images for individual characters from that form as described in “A Universal Way to Collect and Process Handwritten Data For Any Language.” And for Ekush and Bangla lekha isolated we downloaded the datasets from the their respective websites.

3.2.2 Data Preprocessing

For the data preprocessing related task we mainly relied on opencv, numpy and little bit of dask. At first we derived the pixel wise output from each images using opencv, resized them to 28*28 size,applied thresholding, binarized the images and then we stored the flattened vector for all the images into a huge numpy array.

In regard we have to say that since the size of the datasets for Ekush and Bangla isolated were very big so after converting them into numpy array we stored them into a .npy file format for easier and faster access.

Note: Throughout our work we have trained the model on Ekush dataset and tested on Bangla isolated and our own primary dataset separately.

3.3 Machine Learning Model Training, Classification & Evaluation

This step consists of two parts one is the Model training and the other one is the Evaluation Metrics that we have used to assess the performance of our trained model.

3.3.1 Model Training

In this step we have used several models for training and thereafter testing purposes. The models are given below: a)Logistic Regression, b)K-Nearest Neighbors, c)Naïve Bayes, d)Decision Tree, e)Random Forest, f)Support Vector Machines.

a)Logistic Regression: Since here we are dealing with multiclass classification so we have used softmax logistic regression.In this case logistic regression with softmax allows us to predict the probabilities of each class and choose the class with the highest probability as the final prediction.

b)K-Nearest Neighbors: K-Nearest Neighbors is a non-parametric classification algorithm which works by finding the k nearest data points in the feature space to the input data point and assigns the majority class among them as the predicted class.

In our case we have used k=7 which has given us the best accuracy for our validation set.

c)Naïve Bayes: Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem with an assumption of independence between features.

d)Decision Tree: Decision Tree is a tree-like structure where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome.

It recursively partitions the data into subsets based on the most significant attribute at each node.For our purpose we found the sample split to be optimum based on our validation accuracy.

e)Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs most frequently occurring classes as prediction.

In our case we have used n estimators=380 and min sample split=5 which had given the best accuracy in our validation set.

f)Support Vector Machines: SVM is a versatile algorithm that can handle multi-class classification tasks effectively, which works by finding the hyperplane that maximizes the margin between classes. Here since our dataset is quite complex so we have used the nonlinear kernel of SVM in our case which are Polynomial and Radial Basis Function Kernel respectively.

3.3.2 Evaluation Metrics

For evaluation purposes of the model performance we have used Precision, Recall and F1 score as our assessment metrics which are by the way standard evaluation metrics in Machine Learning. To find out these metrics we need True Positive(TP),True Negative(TN),False Positive(FP),False Negative which are key indicators of models performance. For instance, if a classification model predicts words A and B and successfully minimizes errors for both, it demonstrates high precision. Conversely, if the model accurately identifies A as B without any errors, it showcases high recall. However, when a model excels in predicting one class but struggles with another, relying solely on precision or recall can lead to misleading conclusions.

Moreover since there are 50 classes so we have taken the average of recall,precisionand F1 score as our evaluation metrics.

4 Experimental result

4.1 Experimental Settings

All experiment codes were written in Python programming language, version 3.12. Scikit-learn,Opencv,Numpy libraries were used for system training, data processing, and image processing-related tasks. A Computer with an AMD Ryzen 73700x8-core processor , NVIDIA GeForce GT 710 and 16 GB RAM is used as training and testing hardware with Ubuntu operating system.

4.2 Experimental Results and Analysis

The findings of this experiment are based on the generated outputs. The predictions are determined by the count of accurately predicted test images. After the processing and training of our datasets utilizing various ML models in Python, the outcomes are obtained. A prediction is deemed correct only when the predicted value matches with the genuine label of the image. To measure the accuracy, we assessed the number of test images accurately predicted. When the anticipated value coincides with the genuine label of the image, it signifies a correct prediction. Consequently, a variable count is incremented for each correctly predicted image in correspondence to the genuine label. Upon establishing the final count value, it is divided by the total number of input images and then multiplied by 100 to derive the system's accuracy percentage.

As stated previously, throughout our study, we have trained and validated on the data provided by Ekush[7]. Since we have tested our model on two separate datasets, one being our own collected primary dataset and the other Bangla Lekha Isolated[4], this section will be divided into two subsections.

4.2.1 Results on Bangla Lekha Isolated

Model Name	Accuracy	Average Recall	Average Precision	Average F1 Score
Logistic Regression	43.28	44.65	45.64	45.07
KNN	58.64	57.61	61.55	58.11
Naive Bayes	38.04	37.65	37.82	37.73
Decision Tree	24.12	24.88	25.47	25.17
Random Forest	51.74	52.26	57.44	54.64
SVM(Polynomial Kernel)	72.34	73.23	74.19	73.7
SVM(RBF Kernel)	72.81	74.62	74.76	74.69

Table 1: Experimental Results Summary on Bangla Isolated

From Table-1 it is clear that we got highest accuracy on Bangla Lekha Isolated for Support Vector Machine with Gaussian Kernel which is 72.81 percent with an average recall,average precision and average F1 score of 74.62,74.76 and 74.69 respectively.

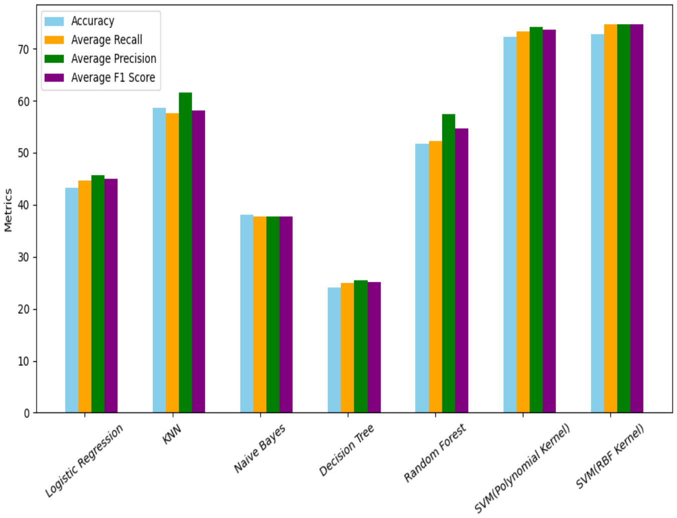


Figure 3: Graphical Representation of Performence on Bangla Lekha Isolated

4.2.2 Results on Primary Dataset

Model Name	Accuracy	Average Recall	Average Precision	Average F1 Score
Logistic Regression	55.68	56.98	58.22	57.62
KNN	62.04	65.71	72.92	68.92
Naive Bayes	47.28	45.92	54.27	49.72
Decision Tree	25.48	23.6	25.66	24.57
Random Forest	63.36	68.7	69.94	69.29
SVM(Polynomial Kernel)	86.8	87.92	89.09	88.48
SVM(RBF Kernel)	88.12	88.28	89.41	88.84

Table 2: Experimental Results Summary on primary dataset

From Table-2 we can see that SVM with Gaussian Kernel gave us the highest accuracy on our own collected primary dataset with classification accuracy of 88.12 percent along with average recall,average precision and average F1 score of 88.28,89.41 and 88.84 percent respectively.

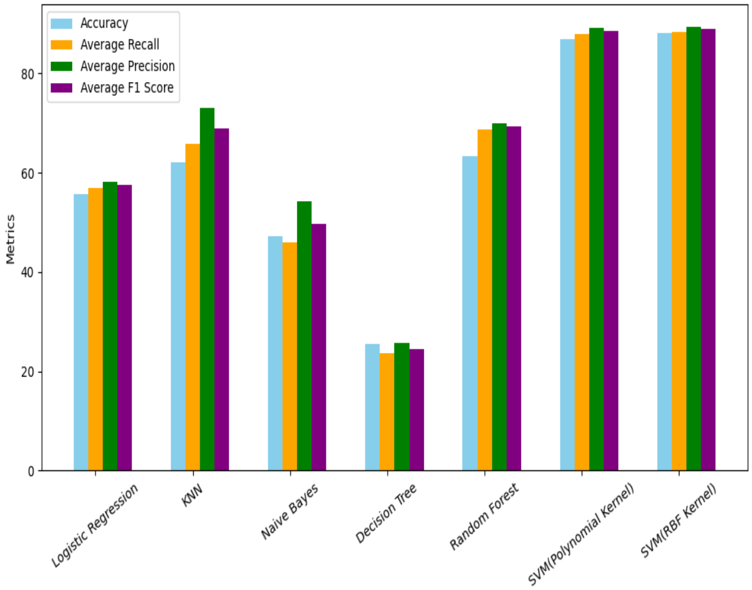


Figure 4: Graphical Representation of Performence on primary dataset

4.3 Comparison with other's work

As we have seen in the previous section that using pixel based approach ,among all the classical Machine Learning models SVM with Radial Basis Function Kernel gives us the maximum accuracy along with all other metrics that we have stated before.Now here we will be comparing our work with work done by others. Table-3 showing a comparison between some of the previous work.

Work	Accuracy
A machine learning approach for Bengali handwritten vowel character recognition[1]	91
BornoNet[5]	95.71
Handwritten Bangla Character Recognition Using Neural Network[2]	84
Bengali handwritten character recognition using Modified syntactic method[3]	95

Proposed Method	72.81(Bangla Isolated)
	88.12(Our Dataset)

Table 3: Result Comparison with others work

5 Summary

Through this project we wanted to show the feasibility of using classical Machine Learning models to classify handwritten bengali characters, and from our experience while working on it we are delighted to see that the work and the initiative we had taken has not been in vain.Because even though our proposed method achieved highest accuracy of 72.81 percent on Bangla isloated and 88.12 percent on our own primary datset but we think through better feature extraction techniques we can increase this percentage a lot,which will definitely be a part of our future endeavours.

References

[1] Shahrukh Ahsan, Shah Tarik Nawaz, Talha Bin Sarwar, M. Saef Ullah Miah, and Abhijit Bhowmik. A machine learning approach for bengali handwritten vowel character recognition. *IAES International Journal of Artificial Intelligence*, 11(3):1143 – 1152, 2022.

[2] Md. Alamgir Badsha, Md. Akkas Ali, Dr. Kaushik Deb, and Md. Nuruzzaman Bhuiyan. Handwritten bangla character recognition using neural network. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2, 2012.

[3] Mohammad Badiul Islam, Mollah Masum Billah Azadi, Abdur Rahman, and M.M.A Hashem. Bengali handwritten character recognition using modified syntactic method. 2005.

[4] Nabeel Mohammed, Sifat Momen, Anowarul Abedin, Mithun Biswas, Rafiqul Islam, Gautam Shom, and Md. Shopon. Bangla lekha isolated. `https://data.mendeley.com/datasets/hf6sf8zrkc/2`, 2017.

[5] Akm Shahariar Azad Rabby, Sadeka Haque, Sanzidul Islam (Md.), Sheikh Abujar, and Syed Akhter Hossain. Bornonet: Bangla handwritten characters recognition using convolutional neural network. *Procedia Computer Science*, 143:528 – 535, 2018.

[6] Akm Shahariar Azad Rabby, Sadeka Haque, Sanzidul Islam (Md.), Sheikh Abujar, and Syed Akhter Hossain. A universal way to collect and process handwritten data for any language. *Procedia Computer Science*, 143:502 – 509, 2018.

[7] Akm Shahariar Azad Rabby, Sadeka Haque, Sanzidul Islam (Md.), Sheikh Abujar, and Syed Akhter Hossain. Ekush a purpose and multitype comprehensive database for online off-line bangla handwritten characters. `https://rabby.dev/ekush/#external`, 2019.

[8] A. B. M. ASHIKUR RAHMAN, MD. BAKHTIAR HASAN, SABBIR AHMED, TASNIM AHMED, MD. HAMJAJUL ASHMAFEE, MOHAMMAD RIDWAN KABIR, , and MD. HASANUL KABIR. Two decades of bengali handwritten digit recognition: A survey. *Machine Learning*, 10, 2022.