Bangla Handwritten Numeral Recognition Using Directional Pattern

Sirajus Salekin
Islamic University
of Technology
Email: salekin@iut-dhaka.edu

Al Shahriar Rubel
Islamic University
of Technology
Email: alshahriar@iut-dhaka.edu

Talha Ibn Aziz
Islamic University
of Technology
Email: talhaibnaziz@iut-dhaka.edu

***Abstract*—Handwritten character recognition has become a challenging and interesting field in the recent days. It has applications in many fields including reading documents, in hand-held notebooks, touch screens, optical scanning by intelligent systems etc. And thus a lot of research is already done and underway on English alphabets and numerals. On the other hand Bangla being the 5th largest spoken language, has not undergone much research. So we have done a directional pattern approach for feature extraction on Bangla numerals and attained a high accuracy of recognition.**

We used LDP(Local Directional Pattern) and GDP(Gradient Directional Pattern) as image pre-processing methods for feature extraction and then the well-known machine learning algorithms KNN and SVM as the classifiers. Then we combined the results to correctly identify the numeral. We used the database CMATERdb 3.1.1 performing a 6-fold cross validation on the 6000 images present to acheive an astounding recognition rate of accuracy 95.62% without over-fitting the data.

***Index Terms*—directional approach, GDP, LDP, KNN, SVM, cross-validation**

I. Introduction

Handwriting recognition is a process where a machine tries to interpret images of characters written by hand into what it represents like a numeral or an alphabet. Like in our case to identify which digit of the bangla numerals is in the image. The image might be preprocessed to remove noise and make it fit for feature extraction. Then there are various feature extraction algorithms to bring out the best features for classifiers to work on. And there are well-known classifiers which work on these attained data or features to identify which is which. This whole process encompasses handwritten character recognition.

A. RECENT WORKS

Even though Bangla Handwritten Recognition has not been extensively studied as English Characters, there still are quite a few attempts of classification. And of them a lot are very impressive. Most of the papers earlier lacked databases so self-written letters or images taken of these characters were taken which lacked the authenticity and varation that a standard database possessed. But that deficit is now not present anymore because there are a few standard and well known databases. Some of them are CMATERdb 3.1.1[1] and ISI[5] which were created respectively in the years 2009 and 2006. We used the former. Therefore there are quite a few noteworthy papers after 2006. For U. Pal, T. Wakabayashi and F. Kimura in 2007[6] used gradient feature to recognize Bengali compound characters. They didn't use any standard database but gained an accuracy of 85.90% which is quite high for compound characters using 5-fold cross validation. M. Zahid Hossain, M. Ashraful Amin and Hong Yan in 2011[7] used rapid feature extraction along with different classifiers and gained an accuracy of 94.12%.They collected data from 120 writers forming a dataset of 12000 images. Mostly the best classifiers used were Convolution Neural Networks and they gave the most accuracies. In 2016[8] M. A. H. Akhand, Mahtab Ahmed and M. M. Rahman performed classification in a mixed dataset of Bengali and English numerals using standard databases. The Bangla digits from CPVR, ISI[5] database gave a maximum accuracy of 98.40% for test sets. Then using autoencoder along with deep convolution neural networking M. Shopon, N. Mohammed and M. A. Abedin in 2016[9] gave an excellent accuracy of 99.50% using both CMATERDB 3.1.1[1] and ISI[5] databases which was the highest accuracy so far. Then in 2017[10] they again performed image augmentation on the CMATER database and gained the accuracy of 99.83%. More work is being done on Bangla Handwritten character recogntion as days pass.

II. DIRECTIONAL PATTERN

Directional Pattern is as the name indicates, it is a pre-processing algorithm to capture the changes surrounding a specific pixel of an image. Thus the changed value does not only depend it's own data but also that of it's adjacent pixels. It rearranges the values of the raw data and assigns weighted values to each previous location using variances in the data surrounding it. For example it uses different masks or kernels to change the values, and these masks are different for the different directions it focuses on to capture the change in variance. Hence the name 'directional pattern'.

We have used two algorithms of dirctional pattern in our method for recognizing bengali numerals efficiently. These are LDP and GDP.

A. LDP (Local Directional Pattern)

The LDP method divides the image into various zones and extracts a histogram from each of these zones which are then used for classification. There are different masks that can be

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\text{East } M_0 \quad \text{North East } M_1 \quad \text{North } M_2 \quad \text{North West } M_3$$

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$\text{West } M_4 \quad \text{South West } M_5 \quad \text{South } M_6 \quad \text{South East } M_7$$

Fig. 1: 8 Kirsch masks for each direction

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

Fig. 2: Sobel Kernels

used to record the changes in the data. The masks that we have used are called 'Kirsch' edge masks. These are 8 masks for eight directions that are applied to the image pixel by pixel form left to right for each line in a zone from to top. We can see from the Figure 1 that each mask is named according to the direction it focuses on and that each direction has different values on the sides depending on what the mask is doing. The centre value is only 0. Next for each of the pixels the values gained from applying the mask gives us a matrix of 8 values for each mask which are placed in the 8 directions except the centre. This matrix is then used to assign a binary digit to each of these values. The attained values are sorted and the first k values are assigned 1 and the rest are assigned 0. The value of k can be selected as per need of performance. Using this binary pattern, the number that is formed is placed in the centre. This number is gained by placing each digit serially in the positions they indicate for example - M_0 is considered for the LSB or the least significant bit and M_7 for the MSB or the most significant bit as shown below. In this way the changed image is attained.

All the 8 masks applied and a new matrix formed.

35	123	79
16	10	201
101	56	99

$\Longrightarrow$mask$\Longrightarrow$

-738	-234	1094
-914	X	902
-746	-82	718

The attained values form the center value.

M_3	M_2	M_1
M_4	X	M_0
M_5	M_6	M_7

$\Longrightarrow$

0	0	1
0	X	1
0	0	1

LDP value = 2^0+2^1+0+0+0+0+0+2^7 = 131

Thus the center value is replaced by the new value.

35	123	79
16	131	201
101	56	99

Like this all the values are replaced one by one to finish the pre-processing.

LDP is a robust process which reduces noise and takes variances of data into consideration. Which is why we used LDP as one of the feature extraction processes.

B. GDP (Gradient Directional Pattern)

GDP is another directional pattern which we used. This is almost the same as LDP except that it works with the gradients of the pixels rather than the grayscale values. In this case before finding the value for the central pixel different kernels are used. The kernel that we used are called 'Sobel' kernels. There are two kernels in 'Sobel' as shown in the Figure 2. Here also we have divided the image into various regions. We got two different matrices from applying these two kernels, one along x-axis and another along the y-axis. Then from the two matrices we get the x and y values for each pixel to calculate the gradient and the magnitudes. In our method as we work with direction we have used only the gradient and not the magnitude.

$$magnitude^2 = G_x^2 + G_y^2$$

$$gradient = tan^{-1}(G_y/G_x)$$

Then we have compared the difference of the gradient values with it's adjacent pixel values and placed a 0 if it is greater to a threshold t and 1 if not. This threshold is to be found out by experimenting. The value is usually from 20^0 to 50^0. Then we divide the image into regions like before and form a histogram for each region. These histograms are the data to be used for classifiers.

GDP is also a pattern which reduces noise like LDP but it is a better feature extraction method than LDP as it works with the gradient rather than just raw values.

III. PROPOSED METHOD

Our proposed method is to use both LDP and GDP to get different vectors. Then use KNN and SVM classifiers on the histograms we obtained and then combine the results like an ensemble method. The 2 steps of our proposed method are briefly explained.

A. Pre-Processing

We did not perform any noise reduction or any other sort of pre-processing on the images rather we directly used directional pattern feature extraction methods on the images. Even without noise cancellation and slant correction etc. pre-processing methods our proposed method performs excellently giving a high accuracy. We did not normalize the images as we used the whole 32×32 bitmap image as the raw data and input for our methods. We used 2 feature extraction methods - LDP and GDP. The LDP method works on the image to give us a histogram of length 256 for each zone or region each containing the count of the pixel values it represents. Then these histograms are placed one after another to create a sample vector for classification.

$$sizeofvector = 256 \times numberofzones$$

In case of LDP the zone of 4×4 gave the best result. And the value of k for sorting was 3 which we used because it gave us most efficiency. The GDP algorithm we used discards the sides of the image so in our case an image of 32×32 will yield a pre-processed image of 30×30. For GDP the best result was given by the region division of 6×6. and for a threshold of 27.5.

B. Classification

We used two kinds of classifiers to get more accuracy. KNN or K-Nearest Neighbors and SVM or Support Vector Machines for multiple classes. We used KNN and SVM on an image after performing LDP and KNN on the same image after performing GDP.

The KNN considers each value of the vector as a co-ordinate and performs euclidean distance calculation to classify each test sample with respect to each training sample. The value of K=3 for KNN gave best classification. And we used SVM to train each class with respect to every other class using the samples and then we all these classifiers to find the class of each test case. Like this we got 3 different results for the same image. If more than one of the three methods gave the same results then we took that as the final result, otherwise we took the classifier with the most accuracy as the final result which was GDP+KNN.

IV. Experimental Analysis and Discussion

A. Dataset Creation

We used the Database CMATERdb 3.1.1 [1] which contains 6000 images of bengali numerals. This means for each digit there are 600 images each being bitmap images of size 32×32. These images are converted into grayscale images and are then pre-processed.

We used k-fold cross validation on the 6000 datasets to create 6 folds each of size 1000. And each time 1 fold acts as test set and the rest 5 folds act as training set. Therefore we focused more on avoiding over-fitting than on accuracy so that the classifier would be more universal, but this decreased the accuracy than some separate datasets created arbitrarily. The highest accuracy attained for the separate folds was 96% when GDP with t=28.5% for 6×6 zones was applied on the third fold.

B. Result Analysis and Comparison

We have performed KNN on different zone configurations of GDP and LDP. KNN yields better results on the image when the value of k is odd and the best results are obtained from k=3.

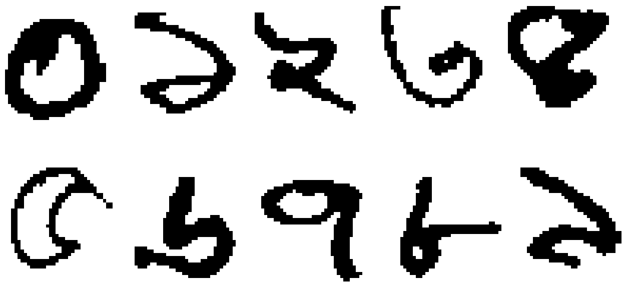
Fig. 3: bitmap images of the benagli digits from CMATERdb 3.1.1

As we have shown in Table I the best accuracy is given by LDP performed on region division of 4×4 which is 93.42%. And for k=1 and k=5 also give good rates of recognition compared to 2 and 4.

TABLE I: Zone-wise accuracy of LDP for KNN

Classifier	LDP(2×2)	LDP(4×4)	LDP(8×8)
KNN (k=5)	88.3333	93.2167	92.9833
KNN (k=4)	87.5500	92.7333	92.7333
KNN (k=3)	88.7500	93.4167	93.2833
KNN (k=2)	85.2167	91.2500	91.6667
KNN (k=1)	88.3333	93.2167	92.9833

The 2×2 gives an image with 4 regions each giving a histogram of size 256 thus the whole image giving a feature vector of dimensions 4×256. Similarly the next two region divisions give feature vectors of sizes respectively 16×256 and 64×254. When SVM is performed on the same image after LDP is done, the accuracies are 92.85% and 92.4333% respectively for zones 4×4 and 8×8.

TABLE II: Zone-wise accuracy of GDP(t=28.5) for KNN

Classifier	GDP(5×5)	GDP(6×6)	GDP(10×10)
KNN (k=5)	93.17	93.90	93.88
KNN (k=4)	91.53	93.32	93.17
KNN (k=3)	93.43	94.05	93.88
KNN (k=2)	92.52	92.22	91.98
KNN (k=1)	93.18	93.75	93.52

In Table II we have shown the results of KNN with different values of k performed after GDP is applied with t=28.5 for different zones. The odd values of k give better results than even ones because odd number of results have a better probability of forming a majority while even ones can result in ties and thus decrease classification accuracy. And the zones of 6×6 give better recognition rate than all the corresponding zones except for k=4 in which case 5×5 is more accurate then 6×6.

We have shown the performance of GDP and LDP for different zones in Table III. As expected GDP being a better algorithm than LDP provides better classification accuracy.

TABLE III: Overall zone-wise Accuracy

Zones	LDP	Zones	GDP
2×2	88.75	5×5	93.43
4×4	93.42	6×6	94.05
8×8	93.28	10×10	93.88

As we can see more zones and less zones both are reducing the accuracy of the methods although usually more regions give better results. Therefore an optimum number of zones must be selected for the best outcome. In case of LDP the zone is 4×4 and for GDP it is 6×6. They respectively divide the image into 8×8 and 5×5 pixel areas. And they give accuracies of 93.42% and 94.05% respectively when KNN(k=3) is applied.

As a result of combining both LDP and GDP we have got a better accuracy on all the digits. The classifiers used are KNN and SVM together which gives a great accuracy of 95.62% for cross-fold validation. Table IV shows the best results of the different pre-processing methods.

TABLE IV: Comparison of different methods

Recognition Methods	Accuracy
KNN(GDP+LDP) + SVM(LDP)	95.62
KNN(GDP+LDP)	94.43
KNN(GDP)	94.05
KNN(LDP)	93.42
SVM(LDP)	92.85
KNN(Basic LBP)	92.23
SVM(GDP)	90.47

We understood after combining both LDP and GDP that ensemble methods give better results than singular methods. LDP gives an accuracy of 93.42% and GDP gives a better accuracy of 94.05%. When GDP and LDP are combined the accuracy increases to 94.43% which is not much improvement as the results to be combined are derived by the same classifier. For 2 options it is difficult to say which is better. When both gave different results we prioritized GDP over LDP. When SVM is used we get 3 results and a better result as the classifier is different. We also implemented different well-known algorithms like Basic Local Binary Pattern or LBP and used KNN as the classifier. All these methods and their accuracies are shown in Table IV.

All the digits of the bengali numerals are not equally difficult to classify. The rate of recognition depends not only on the algorithms and classifiers also on the confusion between 2 or more classes. The more similar they are the less accurately they can be classified. This means unique classes are easier to classify. Each class has it's unique characteristics but the border between them maybe wide or narrow.

TABLE V: Digit-wise Classification

Digit	LDP	GDP	LDP+GDP
0	98.67	99.00	98.83
1	94.83	94.50	96.00
2	97.50	97.67	98.17
3	92.67	93.50	95.50
4	98.17	97.00	97.83
5	88.33	92.00	94.00
6	86.50	87.17	90.33
7	98.00	97.83	98.33
8	99.00	98.83	99.00
9	88.17	83.00	88.17

From Table V we can understand that the digits 6 and 9 have lesser accuracy than other digits. And again some digits can be accurately classified like 0, 7 and 8. This is because 6 and 9 are more similar to each other and thus create confusion. And on the other hand 0, 7 and 8 are not similar to any other digit. And the methods LDP and GDP each have their own weaknesses and advantages. That is why we have combined both to give a better result and they cover for each others' weaknesses.

V. Conclusion

Directional Pattern has not yet been used for recognizing bengali digits so we proposed the directional approach LDP and GDP and for better results we used both together with the classifiers KNN and SVM. But these are not the best classifiers. Neural Networks and Genetic approaches give better results when identifying handwritten characters. Therefore this is just a prologue. If these methods are used with better classifiers then it might give promising results. So Directional Approach should be worked on to further improve recognition rate of Bengali handwritten numerals.

References

[1] Cheng-Lin Liu and Ching Y. Suen, *"A New Benchmark on the Recognition of Handwritten Bangla and Farsi Numeral Characters"*, Pattern Recognition, Volume 42, Issue 12, December 2009, Pages 3287-3295

[2] Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, Dipak Kumar Basu, *"A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application"*, Applied Soft Computing Volume 12, Issue 5, May 2012, Pages 1592-1606.

[3] Tasnuva Hassan, Haider Adnan Khan, *"Handwritten Bangla Numeral Recognition using Local Binary Pattern"*, 2nd International Conference on Electrical Engineering and Information and Communication Technology (lCEEICT) 2015, Jahangimagar University, Dhaka-1342, Bangladesh, 21-23 May 2015

[4] H. A. Khan, A. A. Helal, and K. l. Ahmed, *"Handwritten bangla digit recognition using sparse representation classifier"* in Informatics, Electronics and Vision (ICIEV), 2014 International Conference on 23-24 May 2014, pp. 1-6.

[5] B.B. Chaudhuri. *"A Complete Handwritten Numeral Database of Bangla A Major Indic Script"*, Guy Lorette, Tenth International Workshop on Frontiers in Handwriting Recognition, Oct 2006, La Baule (France), Suvisoft, 2006. ¡inria-00104486¿

[6] U. Pal, T. Wakabayashi and F. Kimura, *"Handwritten Bangla Compound Character Recognition Using Gradient Feature"*, 10th International Conference on Information Technology (ICIT 2007), Orissa, 2007, pp. 208-213. doi: 10.1109/ICIT.2007.62

[7] M. Z. Hossain, M. A. Amin and H. Yan, *"Rapid feature extraction for Bangla handwritten digit recognition"*, 2011 International Conference on Machine Learning and Cybernetics, Guilin, 2011, pp. 1832-1837. doi: 10.1109/ICMLC.2011.6017001

[8] M. A. H. Akhand, Mahtab Ahmed, M. M. Rahman, (2016). *"Convolutional Neural Network based Handwritten Bengali and Bengali-English Mixed Numeral Recognition"*, I.J. Image, Graphics and Signal Processing (IJIGSP), 8, 40-50.

[9] M. Shopon, N. Mohammed and M. A. Abedin, e*"Bangla handwritten digit recognition using autoencoder and deep convolutional neural network"*, 2016 International Workshop on Computational Intelligence (IWCI), Dhaka, 2016, pp. 64-68. doi: 10.1109/IWCI.2016.7860340

[10] M. Shopon, N. Mohammed and M. A. Abedin, *"Image augmentation by blocky artifact in Deep Convolutional Neural Network for handwritten digit recognition"*, 2017 IEEE International Conference on Imaging, Vision and Pattern Recognition (icIVPR), Dhaka, 2017, pp. 1-6. doi: 10.1109/ICIVPR.2017.7890867