

A Study on the Effects of Combining Different Features for the Recognition of Handwritten Bangla Characters

Halima Begum^{*}, Muhammed Mazharul Islam

Department of Electrical and Electronic Engineering, East West University, Dhaka 1212, Bangladesh

ARTICLE INFO

Article history:

Received: 23 January, 2020

Accepted: 26 February, 2020

Online: 16 March, 2020

Keywords:

Handwritten Bangla character recognition

Artificial neural network

Shadow feature

Longest run feature

Chain code histogram feature

Gabor filter

Feature combinations

ABSTRACT

This paper studies and compares the effectiveness of four different features and their combinations on the recognition accuracy of handwritten Bangla characters. The longest run, chain code histogram, shadow, and Gabor filter based features and their eleven (11) combinations were tested on a standard Bangla database of 15,000 basic handwritten characters to compare their recognition performances. From the experiments performed, it was observed that the combination of the longest run, chain code histogram, and the shadow features (having feature vector sizes of 20, 20, and 16 respectively) produce the highest recognition accuracy of 84.01%. Furthermore, inclusion of a feature with a large vector size compared to the other features in the combination generally dominates the recognition accuracy. In our case, inclusion of the Gabor filter-based features with a vector size of 1024 in the combination produced a recognition accuracy of 69.71%, which is worse than the accuracy obtained using the other three features. The analysis of the results indicates that the combinations of different feature vectors produce better accuracy as long as the sizes of each individual feature vector is comparable with each other in the combination.

1 Introduction

Handwritten character recognition has wide commercial applications, e.g., automatic letter sorting based on postal code, extracting information from bank checks and filled up forms, digitization of old handwritten books and documents etc. However, the similarity of shapes of different characters in a particular language, as well as the high variability in the handwritten scripts by different writers makes it quite a challenging task to correctly identify the characters in any language.

Bangla is ranked as the 6th most spoken language in the world and is spoken by nearly 230 million people. In Bangla, there are 50 basic character classes: 11 vowels and 39 consonants. There are also vowel modifiers, consonant modifiers, and other compound characters. Several characters in Bangla have identical shapes, which are only distinguished by the presence of a dot, a short straight line, a curved line, or by the number of loops, strokes etc. These characteristics have given Bangla its unique but complex nature. For recognition of handwritten characters, the obvious similarity of the characters of different classes can hurt the overall accuracy of a system.

Research on Bangla handwritten character recognition is comparatively new. Broadly speaking, two types of research trends are observed in handwritten character recognition. One is the extraction of features from the scanned character images and

then employing a classifier to identify the characters based on the extracted features. Another recent trend is to use convolutional neural network to perform the task of both feature extraction and classification. Success of the first type of methods largely depend on the careful selection of features, while the second type of methods heavily rely on the design of the network structure and the number as well the variability of sample data used to train the network.

Bhowmik et al. [1] used stroke based features on a database of 25,000 characters with recognition accuracy of 84.33%, while Rahman et al. [2] used a multi stage approach on a database comprised of 20 different samples of 49 categories each, with recognition accuracy of 88.38%. Bag et al. [3] used skeletal concavity/convexity of characters along two directions with a recognition accuracy of 60.6%. Basu et al. [4] used an artificial neural network with multi layer perceptron (MLP) on a database of 10,000 characters. Rahman et al. [5] used a convolutional neural network (CNN) to perform the task of both feature extraction and classification on their own dataset of 20000 samples with a recognition accuracy of 85.96%. Alom et al. [6] applied deep CNN for Bangla handwritten character recognition. Although a few of them used large datasets, but these were not publicly available. Therefore, it was not possible to compare the effectiveness of different features proposed by researchers.

Previously, we worked on the combination of three different

^{*}Halima Begum, East West University, A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka 1212, Bangladesh, +88-09666775577-ext 186, & hab@ewubd.edu

features, namely Longest run features (LR), Chain code histogram features (CH) and features extracted using Gabor filter [7]. As a classifier, artificial neural network with back propagation algorithm was considered. We used a publicly available benchmark database [8] of basic Bangla characters and obtained an recognition accuracy of 76.47% considering the combination of the longest run and chain code histogram features [7].

In this paper, we used the same standard database to study and compare the effectiveness of different combinations of four types of features: shadow feature, chain code histogram feature, longest run feature and feature extracted using Gabor filters, in recognizing Bangla handwritten (isolated) basic characters. The purpose of this research is not only to identify the best combinations of features but also to identify the general condition of feature combinations to yield better recognition accuracy.

2 Brief Description of the Database and Preprocessing of the Character Image

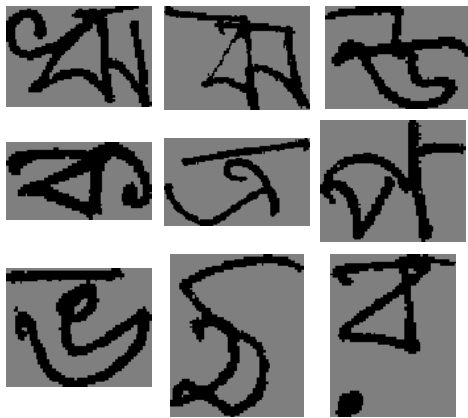


Figure 1: Randomly selected samples from the CMATER training database

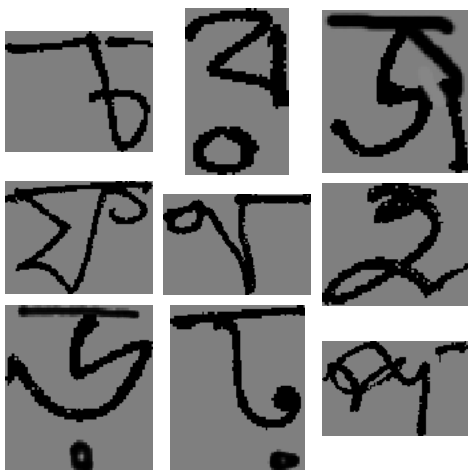


Figure 2: Randomly selected samples from the CMATER test database

In this paper, a benchmark database named ‘CMATERdb 3.1.2’ has been used [8]. The data was collected by the ‘Center for Microprocessor Application for Training Education and Research’ (CMATER), a research laboratory of Jadavpur University, India. The hand written isolated characters were collected from native

Bangla writers of different age, sex and educational groups in formatted data sheets designed by them. The collected data sheets were optically scanned in gray scale with a resolution of 300 dpi. In this way, a total of 15,000 characters of 50 character classes were collected (i.e. 300 character samples for each class). The character images vary from a size of 50 × 17 pixels to 589 × 667 pixels. The images were already divided into a training set and a test set. For any class of character, the number of images in the training set and the test set are 240 and 60 respectively. Fig. 1 shows a few randomly selected samples of Bangla characters from the training database, and Fig. 2 shows samples from the test database.

For recognizing handwritten characters, it is sufficient to work with binary images because the color of the characters do not contribute any additional information in classifying the characters. Binarizing the images also reduce the computational overhead. Furthermore, the size of the images must be uniform across the dataset for proper classification. Therefore, each image was size normalized to 128 × 128 pixels.

Fig. 3 and Fig. 4 show the samples of the preprocessed data. The foreground of the images (i.e. the character strokes) has a binary value of 1, while the background has a value of 0.



Figure 3: The preprocessed samples from the CMATER training database

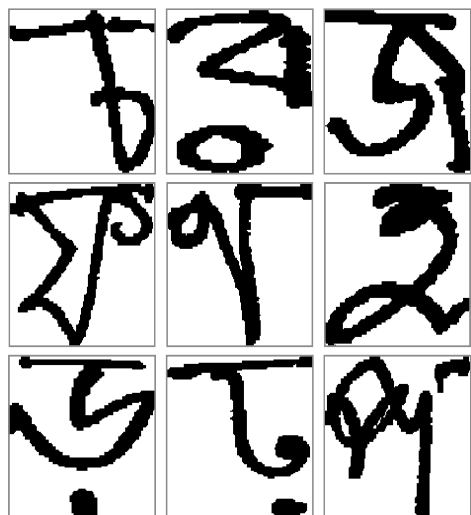


Figure 4: The preprocessed samples from the CMATER test database

3 Description of Features

In order to obtain a good recognition accuracy from a classifier, appropriate features need to be extracted from the character images. Researchers generally look to define features that are unique to the in-class characters but differ significantly between inter-class characters.

In this paper, four features, namely shadow feature, chain code histogram feature, longest run feature, and feature extracted using Gabor filters and their different combinations have been used.

3.1 Shadow Feature

A shadow of a character is the projection of that character in a particular direction (typically on the image borders). The shadow feature computes the length of the shadow in each direction of projection. Projections along the vertical and horizontal directions were considered in this paper [9].

To extract the shadow feature from each of the character images, it was first divided into eight octants as shown in Fig. 5. For each octant, the length of the shadows (of the character segment falling into that octant) were computed on two perpendicular borders. The length of the shadow along a perpendicular border of an octant was divided by the total length of that border to obtain a normalized value and was considered as a feature.

Therefore, a total of 16 shadow features from each character image were obtained. Fig. 6 demonstrates the shadow feature extraction scheme for a sample character (of 16 by 16 pixels), where arrows show the direction of the projection of shadows.

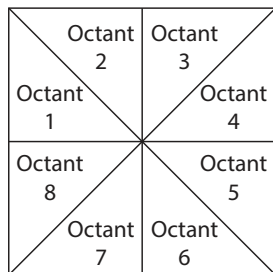


Figure 5: Octants for obtaining shadow features of image

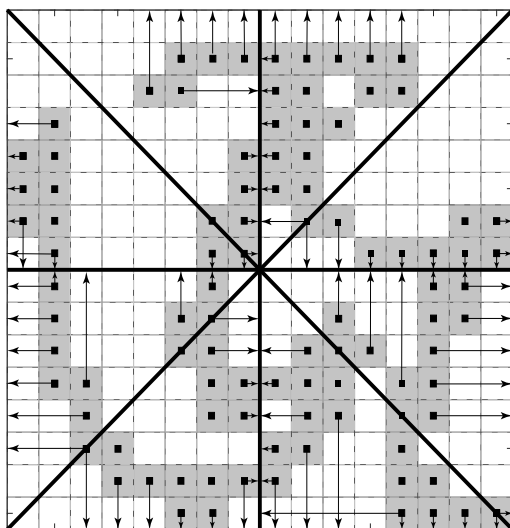


Figure 6: Horizontal and vertical shadow features of image for each octant

3.2 Chain Code Histogram Feature

The chain code carries the information of the shape and the size of a character image. In this method, the directions of movement along the character’s boundary are encoded using a numbering scheme and this allows for a compact representation and reduction of data [10, 11].

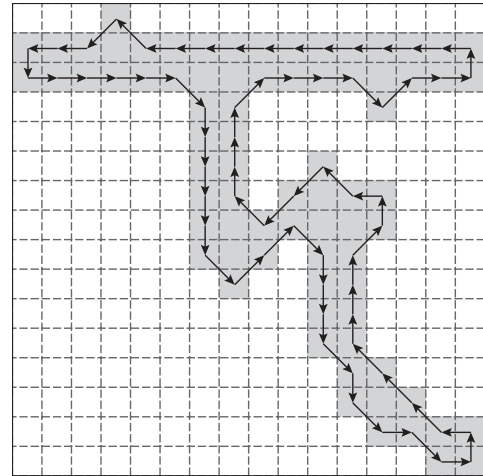


Figure 7: Illustration of the change of direction of boundary pixels for the whole character image

In this paper, to extract this feature, the boundary pixels of the character image (which must be 8-connected or 4-connected) were first detected using Moore’s contour tracing algorithm [12] and then the changes in the direction of the boundary pixels were coded using a scheme proposed by Freeman [13]. Fig. 7 illustrates the change in the direction of the boundary pixels for a sample Bangla character. In the figure, the direction changes along the boundary are denoted with arrows. The change in the direction between two neighboring pixels is coded using Freeman’s chain code. The numbering scheme for the code is illustrated in Fig. 8.

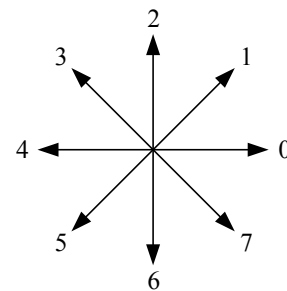


Figure 8: Illustration of the numbering scheme proposed by Freeman’s Chain code considering 8 connectivity of pixels

For example, if the movement from one pixel to the next is along 0°, then the movement is coded as the number 0, if the direction of movement is 45°, then the movement is coded as the number 1 and so on.

Moreover, in the case of character images, the general practice is for the opposite directed movements to be coded with the same value, that is, 0° and 180° are both considered as code 0, 45° and 225° are both coded as 1, 90° and 270° are both coded 2, and 135° and 315° are both coded as 3 [9]. This modification reduces the chain code to four values.

Using the modified chain code, the frequency of occurrence of each directional code (histogram) for a character image was calcu-

lated, and considered as a feature. Therefore, four features were obtained from the entire character corresponding to the histogram of four directions.

In addition to using the whole image for the chain code histogram feature, the character image was further divided into four equally sized sub-images, and then, the chain-code histogram features were also obtained from each of the four sub-images. Fig. 9 shows one of the sub-images (i.e. the top-left section of the character shown in Fig. 7) after dividing into four sub-images.

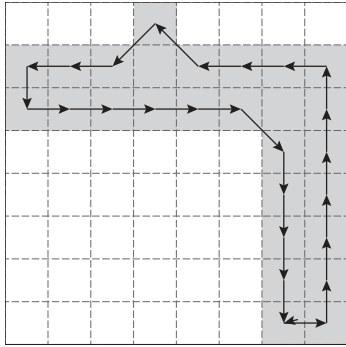


Figure 9: Demonstration of change of direction of boundary pixels of a sub-image

Therefore, using chain code, a total of 20 features (i.e. 4 from the whole image and 16 from the four sub-images) were obtained for each image. The feature vector was then normalized to keep the magnitude of each feature within 0 and 1 [9].

3.3 Longest Run Feature

For a character image, the longest run feature gives a measure of the total longest length of consecutive black (i.e. foreground) pixels along a particular direction. Generally, this feature is extracted for four different directions, i.e. row-wise, column-wise and two major diagonal-wise. Fig. 10 illustrates the row-wise longest run feature calculations for a sample image. From the figure, it is observed that the longest lengths of consecutive black pixels for each of the 16 rows of the character are: 0, 8, 2, 3, 3, 3, 2, 5, 2, 2, 3, 5, 2, 2, 6, and 4 respectively - which add up to 52.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Longest run values
0	0	0	0	0	8	8	8	8	8	8	8	8	8	0	0	0	8
0	0	0	0	2	2	0	0	2	2	0	2	2	0	0	0	0	2
0	1	0	0	0	0	0	0	3	3	3	0	0	0	0	0	0	3
2	2	0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	2	2	0	2	2	0	0	0	0	2	2	2
0	1	0	0	0	0	2	2	0	0	0	5	5	5	5	5	5	5
0	1	0	0	0	0	1	0	0	0	0	0	0	2	2	0	0	2
0	1	0	0	0	2	2	0	0	0	1	0	0	2	2	0	0	2
0	1	0	0	0	2	2	0	0	3	3	3	0	1	0	0	0	3
0	2	2	0	0	0	5	5	5	5	5	0	2	2	0	0	0	5
0	0	1	0	0	0	2	2	0	2	2	0	2	2	0	0	0	2
0	0	2	2	0	0	0	0	2	2	0	0	1	0	0	0	0	2
0	0	0	6	6	6	6	6	6	0	0	0	0	2	2	0	0	6
0	0	0	0	0	2	2	0	0	0	0	0	4	4	4	4	4	4

Figure 10: Longest run values for each row of the character image

From the entire image, 4 longest run features were obtained. The image was then divided into four sub-images around the cen-

ter of gravity of the image [8]. The coordinates of the center of gravity (CG_x, CG_y) of any image were calculated according to the following formula,

$$CG_x = \frac{1}{k} \sum_n x.I(x, y)$$

$$CG_y = \frac{1}{k} \sum_m y.I(x, y) \tag{1}$$

where, x and y are the coordinates of each pixel in the image $I(x, y)$ of size $m \times n$, and k is the count of pixels having $I(x, y) = 1$.

Therefore, a total of 20 features, i.e. 4 from the entire image, and 16 from the four sub-images were obtained altogether. The value of the features so computed is divided by the area of the entire image (or corresponding sub-image) for normalization [8].

3.4 Feature Extracted using Gabor filters

Gabor filters are useful for extracting directional features of a character. A two dimensional (2D) Gabor function is a sinusoidal plane wave modulated by a Gaussian kernel function [14], which is expressed as,

$$g = \exp \left[- \left(\frac{x'^2 + y'^2}{2\sigma^2} \right) \right] \sin \left(2\pi \frac{x'}{\lambda} + \psi \right) \tag{2}$$

where, g is a function of $(x, y, \lambda, \theta, \psi, \sigma, \gamma)$. In Eq. (2), $x' = x \cos \theta + y \sin \theta$, and $y' = -x \sin \theta + y \cos \theta$, where, x, y are spatial coordinates of the image pixels, λ represents the wavelength and ψ is the phase offset of the sinusoid; θ is the spatial orientation of the Gabor function, and σ is the standard deviation of the Gaussian envelope. Fig. 11 shows the Gabor filter kernels for $\theta = 0^\circ, 45^\circ, 90^\circ$ and 135° . Here, the values for $\gamma, \lambda, \psi, \sigma$ are 0.3, 9, 0, and 3 respectively. These represent the optimal values of the parameters and were obtained heuristically.

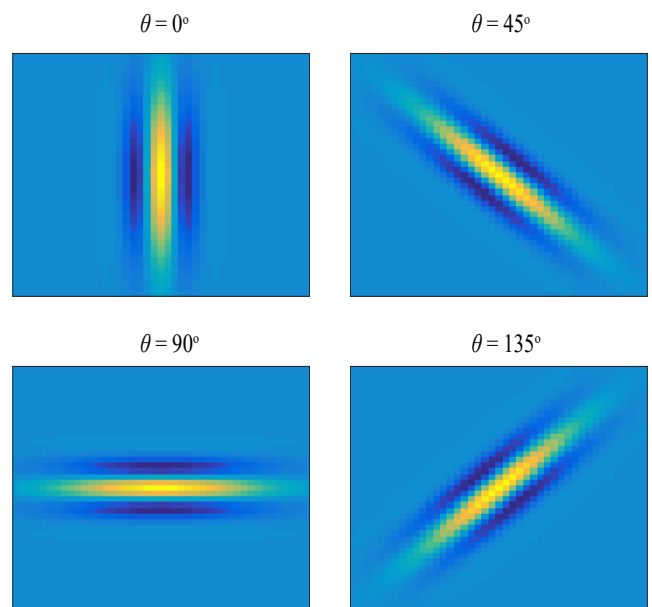


Figure 11: Gabor filter kernels

Fig. 12 shows an original (unfiltered) sample of a character image and Fig. 13 shows the filtered images using the four kernels of the Gabor filter. The enhanced directional features are evident from the filtered images.



Figure 12: Original sample image before applying Gabor filter

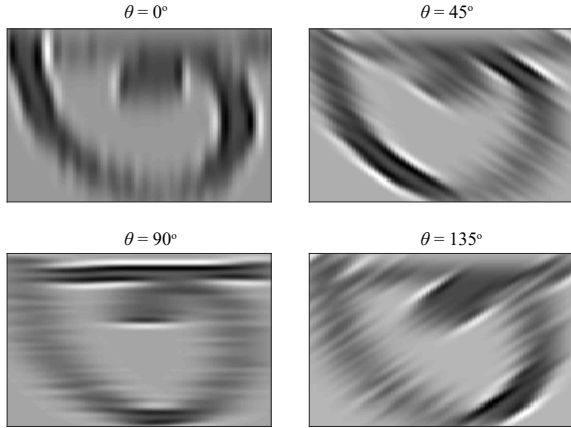


Figure 13: Filtered sample image with enhanced directional feature

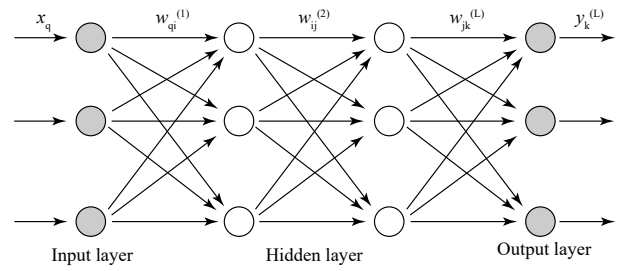


Figure 14: Artificial neural network

Note that, m is the number of classes. In the vectors $\mathbf{d}^{(r)}$, only the correct class element has a value of 1, and all other elements have values of 0. The cost function is defined between the predicted class (\mathbf{y}) and the actual class (\mathbf{d}) over the entire training range as,

$$E = \frac{1}{2} \sum_{k=1}^p \|\mathbf{y}^{(k)} - \mathbf{d}^{(k)}\|^2 \quad (4)$$

During the training phase, the back-propagation learning algorithm is used to optimize the weight values through the minimization of the squared error cost function [16].

As the four filtered images has size 128×128 each, therefore, in this paper, these were down-sampled to get sub-sampled images of size 32×32 each. The four filtered sub-sampled images were then averaged and used as a feature. This was done to reduce the high dimensionality of the feature vector. The feature vector size for each character image is thus $32 \times 32 = 1024$.

4 Artificial Neural Network

An artificial neural network (ANN) was used as the classifier to identify the characters. The ANN consists of nodes in different layers with inter connections [15], where the nodes are distributed in mainly three types of layers, i.e. (i) input layer, (ii) output layer, and (iii) one or more hidden layers. Mathematically, the output of a layer of the neural network can be defined as the weighted sum of n input signals, $x_j = 1, 2, \dots, n$, and the firing of the neurons are controlled by the activation function. The output is expressed as,

$$y = f \left(\sum_{j=1}^n w_j x_j + b \right) \quad (3)$$

where $f(\cdot)$ is the sigmoid activation function, w_j is the weight associated with the j -th input and b is the bias.

We used a feed-forward neural network architecture (with a vanilla structure), where each layer is fully connected to the next as shown in Fig. 14. In the figure, w_{ij}^L is the weight of the connection between the i -th node in the layer $(L - 1)$ to j -th node in the layer L .

To train the network, p -numbers of input-output pairs (called training pairs), defined as $\{(\mathbf{x}^{(1)}, \mathbf{d}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{d}^{(2)}), \dots, (\mathbf{x}^{(p)}, \mathbf{d}^{(p)})\}$ are used. Here, each input $\mathbf{x}^{(r)} = [x_1^{(r)}, x_2^{(r)}, \dots, x_n^{(r)}]^T$ is a feature vector in n -dimensional space, and each corresponding output $\mathbf{d}^{(r)} = [d_1^{(r)}, d_2^{(r)}, \dots, d_m^{(r)}]^T$ (where $d_q^{(r)} \in [0, 1]$) is a vector in m -dimensional space.

5 Classification Results and Analysis

The extracted features from the training data set characters were used as the inputs of the ANN for varying number hidden layers (up to 500), and the recognition accuracy was checked on the test data set. Therefore, the features obtained from the training data actually train the network, and the recognition accuracy indicates how well the trained network is able to recognize the test data set.

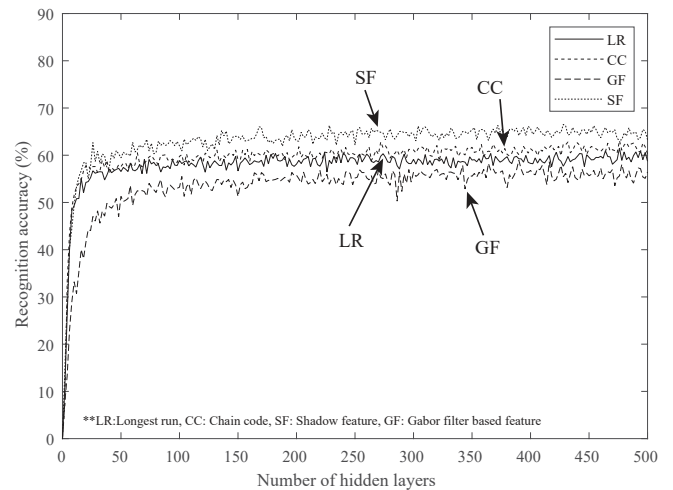


Figure 15: Recognition accuracy of test data for single feature sets

Figure 15 shows the recognition accuracy on the test data against different number of hidden layers, where only single features were used. The recognition performance based on the shadow feature seems to be better than the other three individual features. On the other hand, the recognition performance based on the features extracted using the Gabor filter produces the poorest results among the four.

Figure 16 shows the recognition accuracy on the test data against different number of hidden layers, where different combinations of features (taken two at a time) were used.

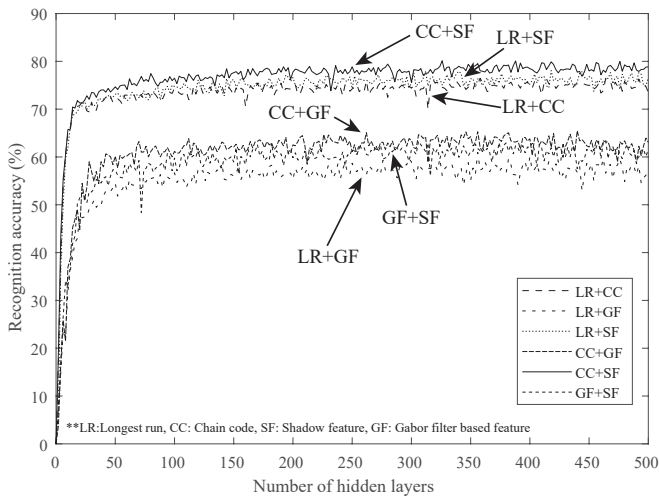


Figure 16: Recognition accuracy of test data for feature sets taken in pairs

Again, the recognition accuracy seems to be poor whenever any combination were used which involved features extracted using the Gabor filter. The combinations (excluding the Gabor filter based features) also produce better results than the those obtained using individual features.

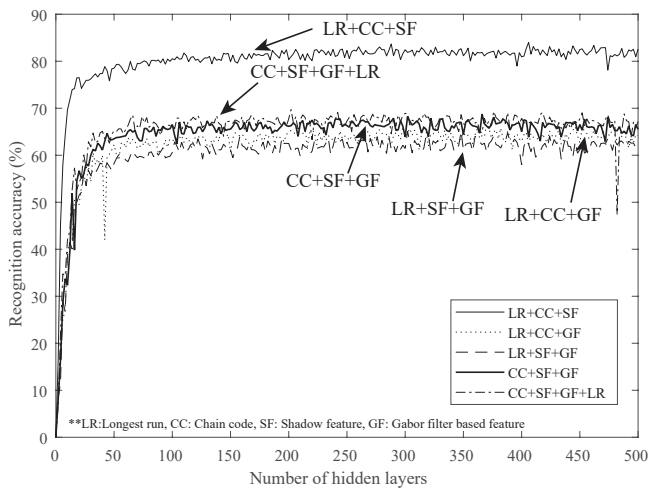


Figure 17: Recognition accuracy of test data for feature sets taken three or more at a time

Figure 17 shows the recognition accuracy on the test data against different number of hidden layers, where different combinations of features (taken more than two at a time) were used. Like the previous case, the recognition accuracy is better without the Gabor filter based features. The recognition accuracy for all the feature combinations (excluding the Gabor filter based features) produce the best results among the combinations.

Table 1 summarizes the maximum recognition accuracy of the neural network (on the test data) under different feature sets and their combinations. The optimal number of hidden layers of the network are also shown in the table.

For the individual feature sets, the highest recognition accuracy is 66.51% using the shadow features, while the recognition accuracy is the lowest (i.e. 59%) for the Gabor filter based feature. It is noted that the shadow feature produces a better result although the feature vector size is only 16, which is the lowest among all the individual feature sets used in this paper, while the

Gabor filter based feature performs poorly, although the size of this feature vector is the largest (i.e. 1024).

Table 1: Classification performance of different feature sets and their combinations

No.	Feature name	Number of features	Classification accuracy	Number of hidden layers
1	Longest run (LR)	20	61.52%	476
2	Chain code histogram (CC)	20	63.47%	476
3	Gabor filter (GF)	1024	59%	428
4	Shadow feature (SF)	16	66.51%	390
5	LR and CC combination	40	76.47%	358
6	LR and GF combination	1044	60.96%	356
7	LR and SF combination	36	78.07%	418
8	CC and GF combination	1044	65.49%	440
9	CC and SF combination	36	80.13%	326
10	GF and SF combination	1040	63.54%	464
11	LR, CC, and SF combination	56	84.01%	406
12	LR, CC, and GF combination	1064	66.68%	370
13	LR, SF, and GF combination	1060	65.11%	348
14	CC, SF, and GF combination	1060	68.77%	366
15	CC, SF, LR and GF combination	1080	69.71%	202

However, higher recognition accuracy was obtained when we used combinations of features on character images. There are six possible combinations of feature sets taken two at a time, four possible combinations of feature sets taken three at a time, and only one combination of all the four feature sets. Looking at the recognition accuracy when using the combinations, it can be seen that a combination of longest run, chain code histogram, and shadow feature produces the best results, i.e. 84.01%. The number of features of this combination was only 56.

It is seen that, any combination of features that contains the Gabor filter based feature produces a comparatively lower accuracy result. The reason for this is, the feature vector size of the Gabor filter based feature is more than 50 times larger than any of the other three individual features, i.e. while the feature vector size of the Gabor filter based feature is 1024, those of the longest run, chain code histogram, and shadow features are 20, 20 and 16 respectively. As a result, the Gabor filter based feature overshadows the other features in these combinations. Therefore, since the Gabor filter based feature itself does not produce a very good recognition accuracy (i.e. only 59%), the recognition accuracy produced by these combinations are weighed down by this dominating sized feature.

Combinations of the other three features (i.e. longest run, chain code histogram, and shadow feature) produce relatively better recognition accuracy, because, the size of the feature vectors of all the three feature sets are comparable. Moreover, for a particular class, some of the character images recognized by each feature sets are non-overlapping. That is, some characters which are *not* recognized using one feature set, are recognized when using another feature set.

Therefore, to get a better recognition accuracy, more than one feature sets are necessary, and the feature vectors of these sets should be comparable in size, otherwise, the results may be dominated by a particular feature set and the purpose of combining different feature sets may not be fulfilled.

6 Conclusion

In this paper, we have compared the effectiveness of four feature sets (shadow, longest run, chain code histogram and Gabor filter based feature) on Bangla handwritten isolated basic character recognition. Moreover, the effectiveness of 11 different combinations of these features were also studied. A benchmark database with 15,000 samples of Bangla basic characters was used for this comparative study.

We observed that, among the individual features, the shadow feature was more effective in Bangla handwritten character recognition compared to the other three features, although its feature vector size was only 16. On the other hand, among the combination of features, the combination of the shadow feature, the longest run feature and the chain code histogram feature yielded the best recognition accuracy, i.e. 84.01%, with a feature vector size of 56 only. In this combination, the size of the feature vectors from the shadow, longest run, and chain code histogram features were 16, 20, and 20 respectively, which are comparable. This implies that, in this combination, the three different features have equal contribution in recognizing the character classes. If the combination includes a feature with a larger vector size (compared to the size of other feature vectors in the combination), then it can strongly influence the recognition accuracy. In our case, when the Gabor filter-based feature (with a vector size of 1024) was added to the combination, it resulted in a comparatively poor performance than those obtained with the combinations of the other features. Therefore, we can surmise that, the combination of features performs better in recognizing character classes as long as the size of the feature vectors are comparable.

In this paper, although we have used hand-coded features, and a feed-forward neural network with backward propagation, but the recent trend is to use convolution neural networks (CNNs). In CNNs, the number of layers is many, and the structure of the network allows for the hidden layers to become sensitized to different features during training. We mainly avoided CNNs because of the small size of the available database, but in our future work, we would like to use CNNs to evaluate the recognition accuracy of handwritten characters and compare with the results obtained in this paper.

References

- [1] T. K. Bhowmik, U. Bhattacharya, S. K. Parui, "Recognition of Bangla Handwritten Characters Using an MLP Classifier Based on Stroke Features" in 11th International Conference, ICONIP 2004, Calcutta, India, 2004. https://doi.org/10.1007/978-3-540-30499-9_125
- [2] A. F. R. Rahman, R. Rahman, and M. C. Fairhurst, "Recognition of handwritten Bengali characters: a novel multistage approach" *Pattern Recognition*, **35**(5), 997-1006, 2002. [https://doi.org/10.1016/s0031-3203\(01\)00089-9](https://doi.org/10.1016/s0031-3203(01)00089-9)
- [3] S. Bag, P. Bhowmick, G. Harit, "Recognition of Bengali Handwritten Characters Using Skeletal Convexity and Dynamic Programming" in 2011 Second International Conference on Emerging Applications of Information Technology, Kolkata, India, 2011. <https://doi.org/10.1109/eait.2011.44>
- [4] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D. K. Basu, "A hierarchical approach to recognition of handwritten Bangla characters" *Pattern Recognition*, **42**(7), 1467-1484, 2009. <https://doi.org/10.1016/j.patcog.2009.01.008>
- [5] M. A. H. Akhand, M. M. Rahman, P. C. Shill, S. Islam, M. M. H. Rahman, "Bangla Handwritten Numeral Recognition using Convolutional Neural Network" in 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 2015. <https://doi.org/10.1109/iceeict.2015.7307467>
- [6] M. Z. Alom, P. Sidike, M. Hasan, T. M. Taha, V. K. Asari, "Handwritten Bangla character recognition using the state-of-the-art deep convolutional neural networks" *Computational Intelligence and Neuroscience*, **2018**, 1-13, 2018. <https://doi.org/10.1155/2018/6747098>
- [7] H. Begum, A. Rafid, M. M. Islam, "Recognition of Bangla Handwritten Characters using Feature Combinations" in 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, India, 2018. <https://doi.org/10.1109/upcon.2018.8597076>
- [8] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, "A benchmark image database of isolated Bangla handwritten compound characters" *International Journal on Document Analysis and Recognition (IJ DAR)*, **17**(4), 413-431, 2014. <https://doi.org/10.1007/s10032-014-0222-y>
- [9] D. R. Birajdar, M. M. Patil, "Recognition of off-line handwritten Devanagari characters using combinational feature extraction" *International Journal of Computer Applications*, **120**(3), 1-4, 2015. <https://doi.org/10.5120/21204-3883>
- [10] Y. Qian, W. Xichang, Z. Huaying, S. Zhen, L. Jiang, "Recognition Method for Handwritten Digits Based on Improved Chain Code Histogram Feature" in Proceedings of 3rd International Conference on Multimedia Technology (ICMT-13), Guangzhou, China, 2013. <http://doi.org/10.2991/icmt-13.2013.53>
- [11] R. C. Gonzalez, R. Woods, *Digital Image Processing*, Prentice Hall, 2008.
- [12] U. Pape, "Implementation and efficiency of Moore-algorithms for the shortest route problem" *Mathematical Programming* **7**(1), 212-222, 1974. <https://doi.org/10.1007/BF01585517>
- [13] H. Freeman, "On the encoding of arbitrary geometric configurations" *IRE Transactions on Electronic Computers*, **EC-10**(2), 260-268, 1961. <https://doi.org/10.1109/tec.1961.5219197>
- [14] D. Gabor, "Theory of communication. Part 1: The Analysis of Information" *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, **93**(26), 429-441, 1946. <https://doi.org/10.1049/ji-3-2.1946.0074>
- [15] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1994.
- [16] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.