

# A Simplified Method for Handwritten Character Recognition from Document Image

Mohammad Imrul Jubair  
Lecturer  
Department of CSE  
BGC Trust University Bangladesh

Prianka Banik  
Lecturer  
Department of CSE  
BGC Trust University Bangladesh

## ABSTRACT

This paper presents a simple and effective technique for converting handwritten textual character from paper document into machine readable form. The proposed method takes the scanned image of the handwritten character from paper document as input and shows the recognized character as its output. Using this method, the object in the converted binary image is segmented and is resized in a global size. After that, morphological thinning operation is applied on that resized object. The image with thinned object is then partitioned into several equal sizes of small cells. A value from each cell is estimated by calculating the proportion of the number of 1 intensity pixels and the number of 0 intensity pixels in the corresponding cell. All of these estimated values are then stored in a one dimensional array. Every element in that array is considered as a single feature value or an attribute for the corresponding image. The k-nearest neighbor classifier is used to classify the handwritten character into the recognized classes of characters. Feature values are estimated from training example images and the classifier is trained using the attributes. After training attribute values for sample image are extracted and passed as inputs in the k-nearest neighbor classifier and the sample image object is grouped using the training dataset into the desired character classes. The proposed technique takes less time to compute, has less complexity and shows desired performance in matching the handwritten characters with the machine readable form and in recognizing them.

## General Terms

Image processing.

## Keywords

Character Recognition, Morphological Thinning Operation, Cell, Feature Value, K-nearest neighbor classifier.

## 1. INTRODUCTION

In the current growing period of technology, optical character recognition (OCR) has become an important field of research. Automatic recognition of printed and handwritten information present on documents like cheques, envelopes, forms, and other manuscripts has a variety of practical and commercial applications in banks, post offices, libraries, and publishing houses. Basically OCR is a mechanism to convert machine printed or handwritten document file into editable text format [1]. The process of handwriting recognition involves extraction of some defined characteristics called features to classify an unknown handwritten character into one of the known classes. A typical handwriting recognition system consists of several steps, as like- preprocessing, segmentation, feature extraction, and classification [2]. Many methods have been proposed for recognizing the handwritten characters such as, HDCRGF [1], IHDCRFDHMM [2], HCRNN [3], EFHSSNNHCR [4], and PABPNN [5] which can recognize the character in image by classifying them, but they take so much

time and the methods are too complex and difficult to implement as well.

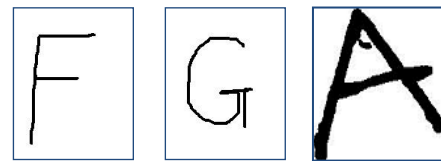


Fig 1: Example of images with handwritten character

In this paper, a simple method for recognizing handwritten character is presented where morphological thinning operation is used. In this approach, character image is brought into a predefined global size and features have been extracted as the attributes of the object. The k-nearest neighbor (k-nn) classifier is implemented in order to classify the image objects. According to this method, the image in any size can be processed, so that no restriction in the size of character object is necessary here. In the next few sections of this paper, different important modules will be discussed which have been used in the proposed method such as morphological thinning operation and K-NN classifier. After that the proposed method will be illustrated with appropriate examples and finally the simulations will be shown elaborately.

## 2. MORPHOLOGICAL THINNING OPERATION

Thinning is a morphological operation that is used to remove selected foreground pixels from binary image which results a single pixel thickness of the binary image. Thinning procedure combines several morphological hit-or-miss transform operations.

### 2.1 The Hit or Miss Transformation

The morphological Hit or Miss Transform is the localization operator in mathematical morphology. It finds occurrences of an object and its nominal surroundings in a set or an image. It is a natural operation to select out pixels that have certain geometric properties, such as corner points, isolated points or border points and performs template matching.

The Hit or Miss Transformation of A by B is denoted by  $A \odot B$ . Here B is a structuring element pair  $B = (B_1, B_2)$ , rather than a single element as before. The hit and miss transformation is defined in terms of these two structuring elements as

$$A \odot B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad (1)$$

Erosion with  $B_1$  determines the location of foreground pixels and erosion of the complement with  $B_2$  determines the location of the background pixels. Performing intersection of these two operations outputs an image which consists of all

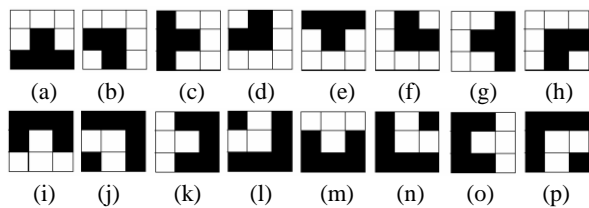
locations that match the pixels in B1 (a hit) and that have none of the pixels in B2 (a miss) [6].

## 2.2 Thinning

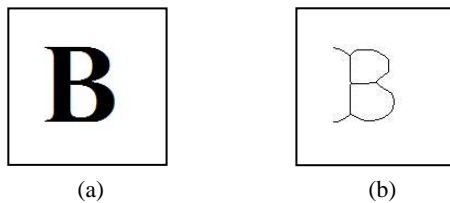
The thinning of an image  $I$  by a structuring element  $J = (J_1, J_2)$  is given by:

$$I \otimes J = I - (I \ominus J) \quad (2)$$

The operation is repeatedly applied until it causes no further changes to the image (i.e., until convergence). The structuring element sequence  $J$  used by us is shown in Fig 2 structuring elements from (a) to (h) show the sequence for  $J_1$  and from (i) to (p) show the sequence for  $J_2$  (complement of  $J_1$ ). The image is thinned using the structuring element pairs  $(J_1^i, J_2^i)$  where  $i=1,2,3,\dots,8$  in sequence. Doing so produces a connected skeleton of the image. The process is repeated in cyclic fashion until the operation produces any further change in the image [6]. Fig 3 shows the example of thinned image.



**Fig 2: Structuring elements.** Here (a) to (h) shows the structuring element sequence  $J_1^1, J_1^2, J_1^3, J_1^4, J_1^5, J_1^6, J_1^7, J_1^8$ . And (i) to (p) shows the structuring element sequence  $J_2^1, J_2^2, J_2^3, J_2^4, J_2^5, J_2^6, J_2^7, J_2^8$ .



**Fig 3: Morphological thinning operation.** Here, (a) is the original image and (b) is the thinned image

## 3. K-NEAREST NEIGHBOR CLASSIFICATION

In pattern recognition, the  $k$ -nearest neighbor algorithm is a method to classify objects based on nearest training sets in the feature space. Using the concept of majority voting of neighbors, an object is classified with being assigned to the class most common amongst its  $k$  nearest neighbors, where  $k$  is a positive integer (typically small). If  $k = 1$ , then the object is simply assigned to the class of its nearest neighbor.

Given a training set  $D$  and a test object  $x = (x', y')$ , the algorithm computes the distance (or similarity) between  $z$  and all the training objects  $(x, y) \in D$  to determine its nearest-neighbor list,  $D_z$ . ( $x$  is the data of a training object, while  $y$  is its class. Likewise,  $x'$  is the data of the test object and  $y'$  is its class). Once the nearest-neighbor list is obtained, the test object is classified based on the majority class of its nearest neighbors:

Majority Voting:  $y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} I(v = y_i) \quad (3)$

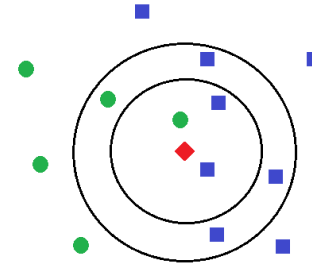
Where  $v$  is a class label,  $y_i$  is the class label for the  $i$ th nearest neighbors, and  $I(\cdot)$  is an indicator function that returns the value 1 if its argument is true and 0 otherwise. The summary of  $k$ -NN classifier can be represented as follow [7]-

**Input:**  $D$ , the set of  $k$  training objects and test object  $z = (x', y')$

**Process:** Compute  $d(x', x)$ , the distance between  $z$  and every object,  $(x, y) \in D$ .

Select,  $D_z \subseteq D$ , the set of  $k$  closest training objects to  $z$ .

**Output:**  $y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} I(v = y_i)$



**Fig 4: An example of  $k$ -NN classification.** The test sample (red object) should be classified either to the first class (blue squares) or to the second class (green circles). If  $k = 3$  it is assigned to the first class because there are 2 square and only 1 circle inside the inner circle. If  $k = 7$  it is assigned also to the first class.

## 4. PROPOSED WORK

Let  $X$  be the input character image with size  $m \times n$ . Generally, documents are prepared by writing on white paper. So, in this paper, we consider the background to be white and the foreground character objects to be black. The following section discusses about the whole procedure. Fig 5 also shows examples of important steps in the recognition procedure.

### 4.1 Extracting Feature-Values

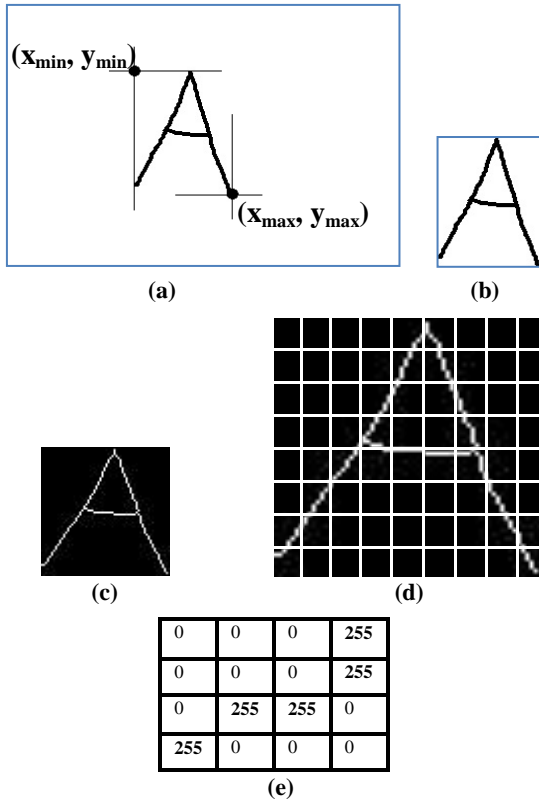
- i.  $X$  is converted into a binary image. So the background pixels will be the intensity of 1 and the foreground object pixels will be the intensity of 0.
- ii. Now, the foreground character object is extracted from the background. For this purpose, coordinates  $(x, y)$  of the black (0) pixels are taken into account and maximum and minimum values are counted among the  $x$  and  $y$  coordinates. From those values object region has been extracted from the background of  $X$ , where  $(x_{\min}, y_{\min})$  will be the upper-left corner's coordinate and  $(x_{\max}, y_{\max})$  will be the lower-right corner's coordinate of the desired object region in  $X$ . Let  $X_E$  be the image matrix containing extracted object region.
- iii. After that,  $X_E$  is resized in a global  $m_r \times n_r$  size. Let  $X_{ER}$  be the resized image.
- iv. Morphological thinning operation is performed on inverted  $X_{ER}$  and a thinned image  $X_{ERT}$  is obtained after the operation.
- v. Now,  $X_{ERT}$  is partitioned into  $N$  number of cells, where each cell has a fixed size of  $s_c \times s_c$ .
- vi. For each cell, proportion of 1's and 0's has been calculated and that value is assigned as the estimated value for the corresponding cell. Suppose cell  $C_1$  has total

$n_w$  number of 1 values and total  $n_b$  number of 0 values. So the proportion is-

$$P_1 = \frac{n_w}{n_b} \quad (4)$$

And thus  $P_1$  is the estimated value for  $C_1$ . Similarly for  $C_2, C_3, C_4, C_5 \dots C_N$  cells the corresponding estimated values  $P_1, P_2, P_3 \dots P_N$  is calculated.

- vii. Estimated  $P_i$  values are then stored in an array  $R_X$  for an individual image, where  $i = 1, 2, 3 \dots N$ .



**Fig 5: (a) shows the extraction of character object from image and (b) is the extracted image, (c) is the result of morphological thinning operation on resized image, (d) shows the concept of partitioning the thinned image in same sized cells and (e) shows an example of calculating estimated value for a cell C where  $n_w = 5$  and  $n_b = 11$ , so  $P_1 = 5/11 = 0.454$**

## 4.2 Classification

The k-nearest neighbor classifier has been used here to classify among different classes or groups of characters where the classifier is at first trained by using some training data sets. For this purpose some images with recognized character are chosen for training and their  $R_X$  arrays are determined using the above method of value estimation described in section 4.1. For an individual training example image  $X_{T1}$ , all the feature values ( $P_{1,1}, P_{1,2}, P_{1,3} \dots P_{1,N}$ ) in its array  $R_{X1}$  will be considered as the attributes for training. Similarly, for another image  $X_{T2}$ , all the feature values ( $P_{2,1}, P_{2,2}, P_{2,3} \dots P_{2,N}$ ) in its array  $R_{X2}$  will be considered as the attributes for training and so on. Let the class label for  $X_{T1}$  be "A" and class label for  $X_{T2}$  be "B" and the next class labels will be "C", "D", "E".....and so on. Suppose a testing sample image  $X_{S1}$  is needed to be classified. For this purpose, array of feature values  $R_{XS1}$  is obtained from  $X_{S1}$  and the array's elements will be the feature values  $Q_{1,1}, Q_{1,2}, Q_{1,3} \dots Q_{1,N}$ .

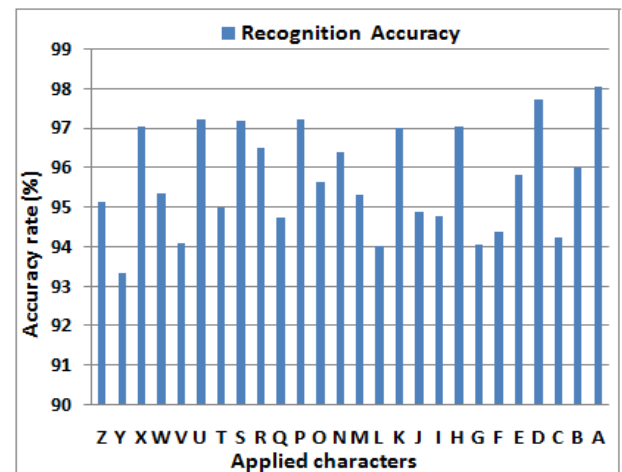
Now, for classification, distances between test sample  $X_S$  and every training objects  $X_{T1}, X_{T2}, \dots X_{TM}$  (where M is the total number of training objects). For an example-

$$\text{Distance}_{\text{Euclidean}}(X_{T1}, X_{S1}) = D_1 = \sqrt{\sum_{j=1}^N (P_{1,j} - Q_{1,j})^2} \quad (5)$$

Similarly, all the distances  $D_2, D_3, \dots D_M$  are calculated in order to find nearest neighbors. To classify and unlabeled test sample, its  $k$  nearest neighbors are identified and the class labels of these nearest neighbors are then used to determine the class label of the testing sample object. After classification, the test sample  $X_{S1}$  will be labeled with any one of the classes of "A", "B", "C", "D", "E".....etc.

## 5. SIMULATION

The proposed method has been simulated using MATLAB programming language. Several images containing character objects of different fonts are used as training example sets. For the simulating purpose, every image is resized in 64 x 64. While partitioning the image in several cells, an image is divided into total 64 cells ( $C_3, C_4, C_5 \dots C_{64}$ ), where every cell's size is 8x8. So there will be total 64 elements of estimated feature value in each  $R_X$  array. That means every object has total 64 numbers of attributes ( $P_1, P_2, P_3 \dots P_{64}$ ). Estimated feature values are stored as attributes for training the classifier. There can be several objects under each class to train. The images used for training can be both handwritten and typewritten. Table 1 shows an example of training dataset by showing the thinned image as well. In this table, the attributes for three labeled classes are shown randomly. Similarly, to check whether the proposed method can recognize handwritten character accurately or not, several images with handwritten characters are tested in simulation. Table 2 shows the feature values (attributes) for some testing sample images. In k-nn classification,  $k=5$  is considered for simulation. Total 780 number of sample images with different handwritten characters collected from different people has been tested using the proposed method and the result shows that the proposed method performs successfully to recognize handwritten characters from document images and its average accuracy rate is 95.688%. Images are tested and the performance of the proposed technique is analyzed. The rate of success in recognizing sample images for different individual characters are shown in Fig 6. Here the accuracy rate is calculated by determining the number of correctly recognizing the unknown handwritten character from document image.



**Fig 6: Accuracy rate (in percentage) of recognizing character for the proposed method**

## 6. CONCLUSION

The method proposed in this paper can recognize handwritten, as well as typewritten characters from digitized document image. In this technique morphological thinning operation is applied on the resized image and it is divided into several cells of same size. After that, for each cell, proportion of the number of 1 dots and 0 dots is calculated. These values are stored in an array for an individual image and are considered as feature attributes. The k-nearest neighbor classifier is used

to assign the unlabeled character object to a labeled class of character. For this purpose, a dataset has been used to train the classifier where the estimated values obtained from each cells are considered as the attributes for the objects. Various simulation results show that the proposed method can perform much accurately to recognize character. Besides, the proposed technique is less complex and easy to implement while recognizing the characters from document image accurately as well.

**Table 1. Example of training dataset.**

Class Label	Training Image	Resized and Thinned	Estimated Feature Values ( $P_i$ ) for Each of the Divided Cells ( $C_i$ )								
			$P_1$	$P_2$	$P_3$	$P_4$	...	$P_{61}$	$P_{62}$	$P_{63}$	$P_{64}$
A			0.00	0.00	0.00	0.00	...	0.032258	0.142857	0.142857	0.122807
			0.00	0.00	0.00	0.00	...	0.00	0.00	0.04918	0.122807
			0.00	0.00	0.00	0.00	...	0.00	0.07	0.00	0.00
B			0.00	0.00	0.00	0.066667	...	0.142857	0.066667	0.00	0.00
			0.142857	0.142857	0.103448	0.103448	...	0.00	0.00	0.00	0.00
			0.103448	0.163636	0.142857	0.142857	...	0.142857	0.142857	0.032258	0.00
M			0.00	0.142857	0.00	0.00	...	0.00	0.00	0.230769	0.00
			0.00	0.00	0.333333	0.00	...	0.00	0.230769	0.00	0.00
			0.142857	0.00	0.00	0.00	...	0.066667	0.00	0.00	0.230769

**Table 2. Example of testing dataset.**

Class Label	Test Image	Resized and Thinned	Estimated Feature Values ( $P_i$ ) for Each of the Divided Cells ( $C_i$ )								
			$P_1$	$P_2$	$P_3$	$P_4$	...	$P_{61}$	$P_{62}$	$P_{63}$	$P_{64}$
Unknown			0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.122807
			0.230769	0.00	0.00	0.00	...	0.00	0.00	0.00	0.230769
			0.00	0.230769	0.142857	0.142857	...	0.142857	0.142857	0.103448	0.00

### **3. REFERENCES**

- [1] Aggarwal, A., Rani, R. and Dhir, R. 2012. Handwritten Devanagari Character Recognition Using Gradient Features. *International Journal of Advanced Research in Computer Science and Software Engineering*, vol.2 - no.5, pp. 85-90.
- [2] Patil, S. B., Sinha, G.R. and Thakur, K. 2012. Isolated Handwritten Devnagri Character Recognition using Fourier Descriptor and HMM. *International Journal of Pure and Applied Sciences and Technology*, vol. 8 – no.1, pp. 69-74.
- [3] Patel, C. I., Patel, R. and Patel, P. 2011. Handwritten Character Recognition using Neural Network. *International Journal of Scientific & Engineering Research*, vol. 2 – no. 5, pp. 1-6.
- [4] Pawar, D. 2012. Extended Fuzzy Hyperline Segment Neural Network for Handwritten Character Recognition. *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, vol. 1-no. IMECS 2012, pp. 43-46.
- [5] Kosbatwar, S. P. and Pathan, S. K. 2012. Pattern Association for character recognition by Back-Propagation algorithm using Neural Network approach, *International Journal of Computer Science & Engineering Survey*, vol. 3 – no. 1, pp. 127-134.
- [6] Bansal, R., Sehgal, P. and Bedi, P. 2010. Effective Morphological Extraction of True Fingerprint Minutiae based on the Hit or Miss Transform. *International Journal of Biometrics and Bioinformatics*, vol. 4 – no.2, pp. 71-85.
- [7] Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J. and Steinberg, D. 2007. *Top 10 algorithms in data mining*. Knowledge and Information Systems, Springer-Verlag New York, Inc, vol.- 14, no. 1, pp. 1-37.
- [8] Charles, P. K., Harish, V. Swathi, M. and Deepthi, CH. 2012. A Review on the Various Techniques used for Optical Character Recognition. *International Journal of Engineering Research and Applications (IJERA)*, vol. 2 – no. 1, pp. 659-662.
- [9] Gonzalez, R.C., and Woods, R.E., 2004. *Digital Image Processing (2nd edition)*, Pearson Education.
- [10] Baxes, G. A. 1994. *Digital Image Processing: Principles and Applications*, John Wiley & Sons, New York.