Numeral Handwritten Hindi/Arabic Numeric Recognition Method

Mohamed H. Ghaleb[1], Loay E. George[2], and Faisel G. Mohammed[3]
Computer of Science, College of Science, University of Baghdad, Iraq

Abstract: Handwritten numerals recognition plays a vital role in postal automation services. The major problem in handwritten recognition is the huge variability and distortions of patterns. The aim of the current research work is to develop a heuristic based method has good recognition efficiency for recognizing numeral free handwritten objects. In this research, the introduced method for extracting features from patterns is based on (i) the percentage of strokes in both horizontal and vertical directions and (ii) some morphological operations. The proposed method gives good recognition result, the attained recognition rate is 98.15%, the number of tested samples was 4500 samples.

Keyword: Artificial intelligence, pattern recognition, image segmentation, character recognition, handwritten recognition**.**

1 INTRODUCTION

Handwritten recognition has attracted many researchers across the world **[1, 9, and 10].** The problem of automatic recognition of handwritten text as opposed to machine printed text is a complex one, especially for cursive based languages. Several researchers have introduced algorithms for character recognition for different languages such as English, Chinese, Japanese, and Arabic **[2, 4, and 5].**

Typical Optical Character Recognition (OCR) system consists of the phases: preprocessing, segmentation, feature extraction, classifications and recognition. The output of each stage is used as the input of next stage. Preprocessing stage consists of many adjustment operations for slant correction, normalization and thicking. Many newly proposed methods have been introduced for the purpose of feature extraction **[3, 5].**

Most of the Indian scripts are distinguished by the presence of matras (or, character modifiers) in addition to main characters, while the English script has no matras. Therefore, the algorithms developed specifically for English are not directly applicable to Indian scripts **[6].**

2 RELATED WORKS

In recent years some researchers have developed computational intelligence models for accurate recognition of Arabic text. Al-Omari **[7]** used an average template matching approach for recognizing Arabic (Indian) numerals. He suggested the use of feature vectors representing a set of significant boundary points distances from the center of gravity (COG) of the numeral object. He were used these features to derive a model for each numeric digit. An overall hit ratio of 87.22% was achieved in the preliminary results. This ratio reached 100% for some of the digits. But there was misinterpretation between similar digits like (6) and (9). Classification was performed using the Euclidean distance between the feature vector of the test samples and the generated models.

Sadri et al. **[8]** proposed the use of support vector machine for the recognition of isolated handwritten Arabic/Persian numerals. The method views each digit from four different directions, and then extracting the features used which are used to train SVM classifiers to recognize the digit classes. An average recognition rate of 94.14% was obtained.

A new method based on Hidden Markov Model (HMM) for recognition of isolated handwritten Arabic (Indian) numerals was presented by Mahmoud **[9]**. In his method, four sets of features (i.e. angle, circle, horizontal and vertical (ACHV)), were generated based on the segmentation of numeral image, and for each segment the ratio of black pixels to segment size was computed. These features were used for training and evaluating the HMM models. Average recognition rate of 97.99% was achieved.

The use of abductive machine learning techniques for the recognition of handwritten Arabic (Indian) numerals was demonstrated by Lawal **[10].** An average recognition rate of 99.03% was achieved with a set of only 32 features based on FCC codes.

3 PROPOSED SYSTEM DESCRIPTION

Like other languages Indian has 10 basic digits, the scope of this paper is limited to develop an approach for detecting the handwritten Hindi numerals (from one to nine: 1-9) which are commonly used in Arabic writing. Each numeral type was written by different peoples in different style, as shown in Table (1). The proposed system has ability to recognize numeral objects in different background and foreground colors as shown in Table (1).

Table 1: Different Styles of Handwritten Hindi Numerals Samples

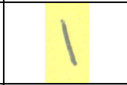
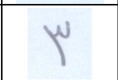
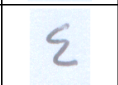
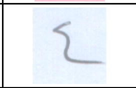
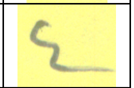
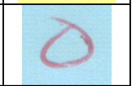
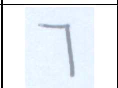
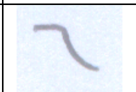
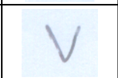
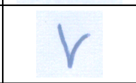
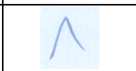
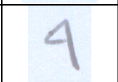
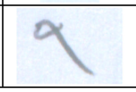
Numeral	Style1	Style2	Style3
1			
2			
3			
4			
5			
6			
7			
8			
9			

Figure (1) shows the scheme of the proposed system. In general the first step in this system is involved with loading the numeral color image file; then it is converted to gray image, and to binary image using thresholding method.

The obtained binary digit object image is enhanced using median filter to remove unwanted isolated pixels (noise). After this step, the image is clipped, and then the numbers of strokes are calculated via horizontal and vertical scans. The counted numbers of strokes are normalized to determine the corresponding percentages of strokes.

The determined percentage is then tested to decide the class/type of tested image. Some morphological criteria are used to discriminate between samples belong to same class.

The stages of the developed system are shown in Figure (1), they are:

1. *Preprocessing and segmentation stage (image to image).*
2. *Features extraction stage (image to feature).*
3. *Decision making stage (feature to interpretation).*

The details of the each stage are clarified in the next sections.

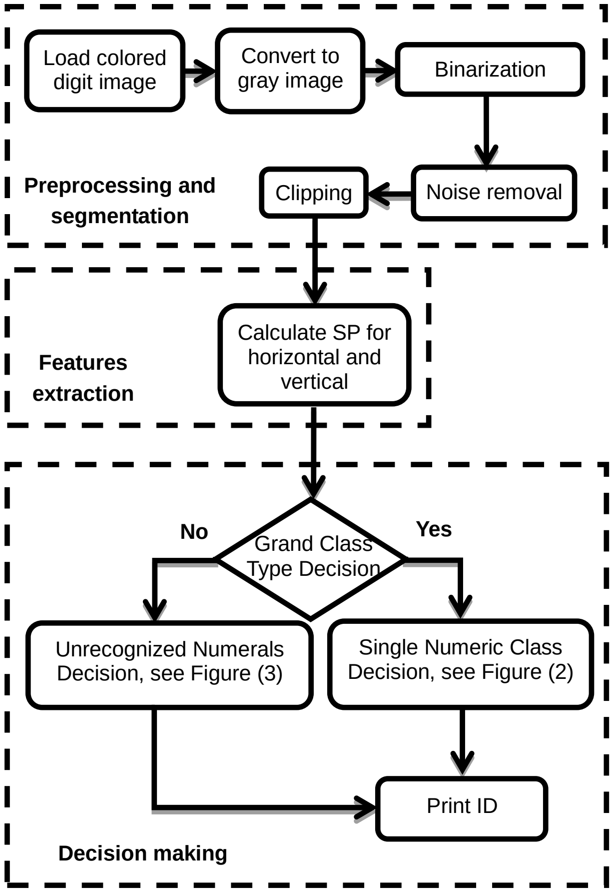


Figure 1: The Block Diagram of Proposed System, SP denotes to strokes percentages

3.1 Preprocessing and Segmentation

This stage consists of the following tasks:

- **A. Load colored digit image**: The system has ability to load bitmap image file format.
- **B. Convert to gray image**: The digit image is converted to gray image.
- **C. Binarization**: In the normal case the numeral image should consist of two colors (i.e., foreground & background), the largest repeated color refers to the background and the second largest repeated color refers to the foreground color. The steps taken to convert grayscale image to binary image are:
  - a. *Determine the histogram of the image.*
  - b. *Search in the histogram to find the largest two peaks they should be separated at least by certain colors. So, the distance between the locations of these two peaks should be more than a predefined minimum distance value. The midpoint between these two peaks is considered as the threshold value used to convert image from gray to binary:*

    Threshold value = (Pe+Ps)/2

    Where, Pe is the value of right peak, and Ps is the left peak value.

c. *Scan all images pixels if the gray pixel value is close to the highest peak value then set pixel value (0) otherwise set the value (1).*
Where black pixel (0) refers to the background and white pixel (1) refers to the foreground.

D. Noise removal: To remove noise from binary digit image the median filter was used.

E. Clipping: this process used to clip the numeral image from input image such that the new image boundaries confine the numeral object area. The scanning process consists of the following steps:

a. *Find the left and right edge (the most left and right columns contain white pixels (1)).*
b. *Find the top row and bottom row (the first and last row contain white pixels (1)).*

The new width and height of the clipped image are:-
Width = right edge – left edge + 1
Height =bottom edge – top edge + 1
So, the size of the clipped image is (width, height)

3.2 FEATURES EXTRACTION

The extracted features are mainly based on the percentage of number of strokes for both horizontal and vertical directions. The minimum number of strokes in both horizontal and vertical is one stroke because the characteristics of Hindi figure (1 to 9) are being connected, and the maximum number of strokes is 4 which can be found in the numeral image "٤". The calculation steps for the percentages of number of strokes for both horizontal and vertical are:

A. *Let HP and VP as array [1 to 4] which represent the horizontal and vertical percentage of the number of found strokes {1, 2, 3, 4}.*
B. *Let Counter as an array [1 to 4] that represents the counters for the four types of strokes. The initial values of the array elements are set zero.*
C. *Scan horizontally each row in the clipped image:*
C1. Set Stroke = number of strokes counted in the tested row
If Stroke=0 then set Stroke=1
Else If Stroke>4 then set Stroke=4
Increment Counter[Stroke]by 1
C2. Set HP[I]=Counter[I]/Width
Where I represents the four strokes, and HP is the horizontal percentage Stroke of I.
D. *Scan vertically each column in the clipped image:*
D1. Initialize Counter array (all values of the array elements are set equal to 0).
D2. Repeat c1 to c2 steps but for columns instead of rows and VP instead of HP

3.3 DECISION MAKING

This stage implies of the following operations:

A. Grand Class Type Decision: This step classified the input images into the following 8 grand classes:

a. *1st GClass (N1): This class contains the image object that recognized as number "١". The features of Class1 are the height and width of clipped image, this means we don't need to calculate the percentages of the four strokes types.*
b. *2nd GClass (N4): This class contains the image object that recognized as number "٤".*
c. *3rd GClass (N5): This class contains the image object that recognized as number "٥".*
d. *4th GClass (N16): This class contains the numeral objects that recognized as number "١" or "٦".*
e. *5th GClass (N78): This class contains the image objects that recognized as number "٧" or "٨".*
f. *6th GClass (N24): This class contains the image objects that recognized as number "٢" or "٤".*
g. *7th GClass (N39): This class contains the image objects that recognized as number "٣" or "٩".*
h. *8th GClass (N-1): This class contains the image objects that are not recognized.*

The following set of criteria was used to recognize the above listed grand classes:

*(i) If Height > Width*4 the Gclass=1*
(ii) If HP(1)>0.7 and (VP(4)>0.05 or VP(3)>0.25) the Gclass=2
(iii) If HP(2)+HP(3)>0.53 and VP(2)+VP(3)>0.53 the Gclass=3
(iv) If HP(1)>0.85 and VP(1)>0.85 the Gclass=4
(v) If HP(2)>0.38 and VP(1)>0.81 the Gclass= 5
(vi) If HP(1)>0.89 and HP(3)<0.02 and VP(2)>0.3 the Gclass=6
(vii) HP(2)+HP(3)>0.2 and HP(2)+HP(3)<0.35 the Gclass=7
(viii) Otherwise Gclass=8

The fixed percentages values mentioned is above list of criteria are used to reduce the interlacing between classes, wing other value will lead to increase the interlacing between classes see table (2). The fixed percentages values have been found by following trail procedure. In case of finding a grand classes consist of more than one numeral (i.e., 4th, 5th, 6th, 7th grand class) then the next stage (i.e., B) should applied.

B. Single Numeric Class Decision

In single class type decision making, their will be four expected grand classes ("16", "78", "24", "39") and each grand class has more than one expected numeral. The sub-classification task was handled using different set of criteria, each set is designed to handle the numerals belong to one of the grand classes. So, four sets of morphological based criteria were introduced the handle within classification task see figure (2):

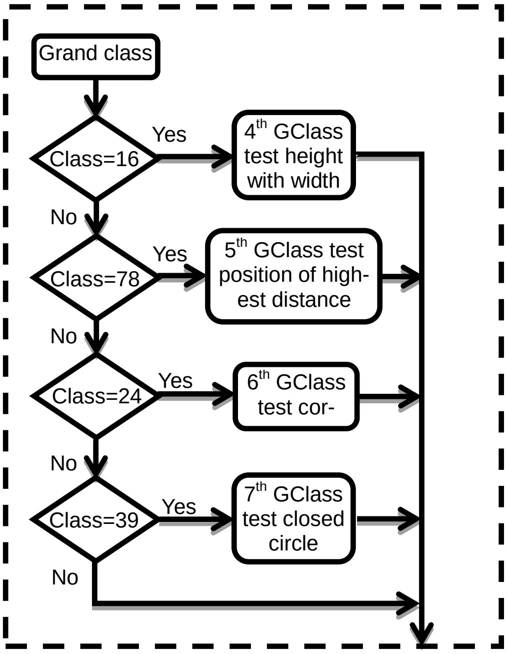


Figure 2: Single Numeric Class Decision

a. **The fourth GClass (N16):** To distinguish between "١" and "٦" numerals from input image nominated as "16" grand class. The width to height ratio of the tested clipped image is used as follows:
   *(i) If height > width *2 then class=1 Else class=6*

b. **The fifth GClass (N78):** To distinguish between "٧" and "٨" numerals from input image nominated as "78" grand class. The main difference between "٧", "٨" is the position of highest distance found between the two strokes during the horizontal scans. If the highest distance is found at the top side of the image then the numeral is recognized as "٧", otherwise it is "٨". The implementation steps are classified in the following:
   - *Scan horizontally each row in the clipped image.*
   - *Set count=0.*
   - *Calculate the distance between first and second strokes for row that contain two strokes only if this distance is more than or equal the distance of the previous scanned row then increment count by 1.*
   - *If (count div number of the rows contain only two strokes) > 0.7 then the tested numeral=7 otherwise it is numeral =8*

c. **The sixth grand class (N24):** To distinguish between "٢" and "٤" numerals from the input image nominated as "24" grand class. The fact that numeral "٤" has only two corners while numeral "٢" has only one corner is taken if we take a horizontal scan on numeral "٢" from left side. Starting from bottom side, we will see that the line direction is coming from right to left and at the upper side it will reverted to be from left to right. While, for numeral "٤" the line direction must iterated from right to left and then from right to left for twice times.

d. **The seventh grand class (N39):** To distinguish between "٣" and "٩" numerals from the input image nominated as "39" grand class, the fact that numeral "٩" have a closed circle in the high part of its image is taken. This circle must be fully closed or, up to some extent. While the numeral "٣" doesn't have this circular segment.

C. Unrecognized Numerals Decision

In this stage the unrecognized numerals through the first (i.e., grand) classifier will distinguished. This could be done through the following recognition steps see figure (3):

a. If a closed circle is found in the high part (around 60% of the image) then the numeral is recognized as "٩".
b. If all image content represent a semi closed circle then the numeral is recognized as "٥".
c. If HP(2) + HP(3) > 0.65 and VP(2) + VP(3) > 0.06 then done step (d).
   Otherwise this image unrecognized as numeral.
d. Test the curvature of the upper part of the numeral image; if its type is convex the numeral is considered "٢" or "٤" (recognized as in sixth grand class (N24)), while if it is a concave type the numeral is considered "٣".

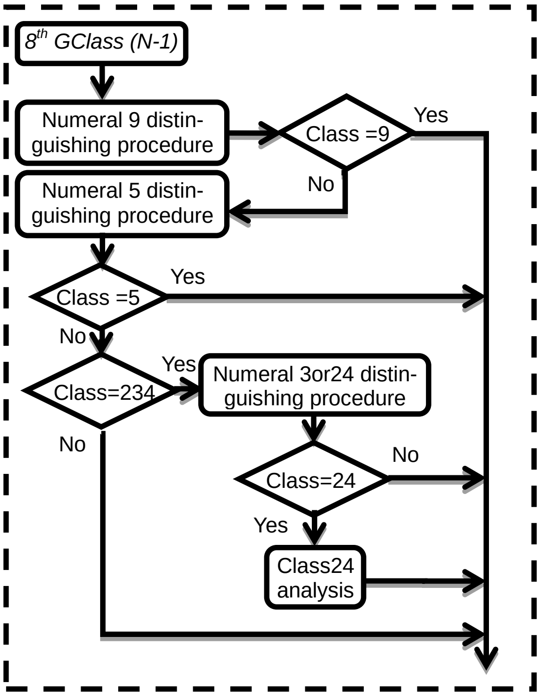


Figure 3: Unrecognized Numerals Decision

C.1 Numeral 9 distinguishing procedure:

This can be done as follows:

a. Let Closed Circle Counter CCC=0 and Two Strokes Counter TSC=0.

b. *Calculate number of strokes in the high part (around 60%) of the image. If the number of counted strokes equal 2 or 3 then increment TSC by 1 otherwise go to step (2).*
c. *If the number of counted strokes=3 then the two strokes that have biggest distance are used.*
d. *Allocate the first point belong to the background after the 1st stroke.*
e. *Allocate the last point belong to the background before the 2nd stroke.*
f. *Set column counter ColC=0.*
g. *Start vertical scan along the points from 1st to 2nd point, at each vertical scan instance if there are upper and lower strokes are found then increment ColC by 1.*
h. *If we reach ColC>60% of the two points distance then increment CCC by 1.*
i. *Search in all columns in image, and repeat steps (b) to (h) (but for columns instead of rows).*
j. *If the attained CCC>10% of total points within the image and CCC>60% of TSC, then the image have a closed circle otherwise it doesn't.*

C.2 Numeral 5 distinguishing procedure:

The same above mentioned steps for numeral "9" are followed but the search will be for whole image region.

4. Results and Conclusions

The conducted tests have been applied on a set consist of (4500) numeral images extracted from (42) scanned documents prepared by (42) persons. The tested results indicated that the attained recognition rate is 98.15%. This recognition rate is achieved when the percentage of strokes for both horizontal and vertical is utilized as a discriminating feature and the interlaced classes are separated using morphological operation, as explain in the section 2. Table (2) shows the attained recognition ratios whine the criteria listed in section 2.3.1 have been used. Table (3) shows the recognition ratios of (success rate, failure rate, and misclassified rate) for all numerals from 1 to 9.

Since the feature based on strokes percentage are reflection invariant (vertical and horizontal direction) so, for this reason some additional morphological attributes (like distance between strokes) have been use to signified between numerals that are mirrors to each other (e.g. the numerals "٧" and "٨").

Table2. The Recognition for the 8 Grand Classes

Class \ NO	1	2	3	4	5	6	7	8	9
1	45								
4				59					
5					89.6				
16	53.8					97.4			
78							97	99.4	
24		89.2		11					
39			63.2						47
-1	0.8	10.8	36.8	30	10.4	2.6	1.8	0.4	52
Misclassified Rate	0.4	0	0	0	0	0	1.2	0.2	1

Table3. The final recognition ratio of (Success Rate, Failure Rate, and Misclassified Rate)

Samples	1	2	3	4	5	6	7	8	9
Success Rate	96.6	99.8	99.8	97.6	99.2	97.4	97	99.4	96.6
Failure Rate	3.4	0.2	0.2	2.4	0.6	1.4	2.8	0.4	3.4
Misclassified Rate	0	0	0	0	0.2	1.2	0.2	0.2	0

REFERENCES

[1] Wadhwa, D. and Verma, K., "Online Handwriting Recognition of Hindi Numerals using Svm", *International Journal of Computer Applications*, Vol. 48 – No. 11, pp (13-17), June 2012.

[2] Elnagar, A., Al-Kharousi, F., and Harous, S., "Recognition of Handwritten Hindi Numerals using Structural Descriptors", *IEEE International Conference*, Vol. 2, pp (983-988), 12-15 Oct 1997.

[3] Sinha, G., Rani, R., and Dhir, R., "Handwritten Gurmukhi Numeral Recognition using Zone-based Hybrid Feature Extraction Techniques", *International Journal of Computer Applications*, Vol. 47– No.21, pp (24-29), June 2012.

[4] Rani, A., Rani, R., and Dhir, R., "Combination of Different Feature Sets and SVM Classifier for Handwritten Gurumukhi Numeral Recognition", *International Journal of Computer Applications*, Vol. 47– No.18, pp (28-33), June 2012.

[5] Dhandra, B.V., Benne R.G., and Hangarge, M., "Printed and Handwritten Kannada Numerals Recognition Using Directional Stroke and Directional Density with KNN", *International Journal of Machine Intelligence*, Vol. 3, Issue 3, pp (121-125), November 2011.

[6] Hanmandlu, M., Nath, A.V., Mishra, A.C., and Madasu, V.K., "Fuzzy Model Based Recognition of Handwritten Hindi Numerals using Bacterial Foraging", *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*, 11-13 July 2007, Melbourne, Australia.

[7] Al-Omari, F., "Hand-Written Indian Numerals Recognition System Using Template Matching Approaches", *ACS/IEEE International Conference on 2001*, pp (83-88), 25-29 Jun 2001.

[8] Sadri, J., Suen, C.Y. and Bui, T.D., "Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits", *Toosi Univ. of Tech*., 2nd MVIP, Vol.1 pp (300-307), Feb 2003, Tehran, Iran.

[9] Mahmoud, S., "Recognition of Writer-Independent off-line Handwritten Arabic (Indian) Numerals using Hidden Markov models", *Signal Processing*, Vol. 88, Issue 4, pp (844-857), April, 2008.

[10] Lawal, I.A., Abdel-Aal, R.E., and Mahmoud, S.A., "Recognition of Handwritten Arabic (Indian) Numerals

Using Freeman's Chain Codes and Abductive Network Classifiers", International Conference on Pattern Recognition, *IEEE Computer Society*, pp (1884-1887), 23-26 Aug. 2010.