A System for Joining and Recognition of Broken Bangla Numerals for Indian Postal Automation

K. Roy, U. Pal and B. B. Chaudhuri
CVPR Unit; Indian Statistical Institute, Kolkata-108; India
umapada@isical.ac.in

Abstract

In this paper, we present a system towards recognition of Bangla pincode numerals for Indian postal automation. In the proposed system, at first, using structural features the broken numerals are joined. Next combining Neural Network (NN) and tree classifier based approach the numerals are recognized. Considering similar shaped numerals at first, NN classifies the 10 numerals into six groups. Next tree classifier is used for final recognition. The features used for the NN based recognition are the number and position of end points, junction points, position of the centre of gravity, and distance between the centre of the bounding box and the centre of gravity etc of a numeral. Different features used for tree classifier are based on water reservoir concept, structural features, and topological features. Overall accuracy of the proposed system is at present 94.21%.

1. Introduction

Postal automation is a topic of interest for research for the last few years and many pieces of published work are available towards automation of non-Indian languages documents [1,2,3]. Some systems are available for English documents of USA, UK and Australia post offices, but no work has been done towards the automation of Indian postal system apart from the work reported in [4].

The paper deals with pin code based automation of postal documents written in Bangla, the second most popular script in Indian subcontinent. One of the important components in postal automation is to recognize pincode numerals. Some of the extracted Bangla numerals are broken (shown in Fig. 1) or contain noise, making it difficult for recognition. Paper and writing quality of the document and poor digitization are the main drawback of the recognition of such documents.

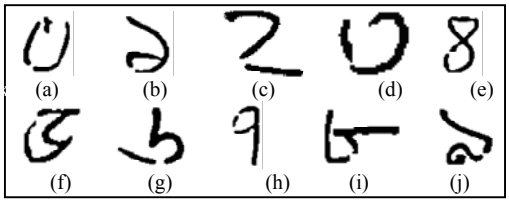

Fig. 1. Examples of some broken numerals. Numeral (a) zero, (b) one, (c) two, (d) three, (e) four, (f) five, (g) six, (h) seven, (i) eight, (j) nine.

In this paper an efficient method is described to analyze possible structures of broken Bangla numerals so that they can be joined and recognized. For this purpose, in Section 2 a set of preprocessing algorithms is described. In Section 3 the reconstruction of broken numerals is elaborated. Feature selection and extractions are explained in Section 4. In Section 5 at first MLP-network based classification is discussed, then tree classifier based final recognition is designed. The experimental results are discussed in Section 6. Finally, the conclusion is given in Section 7.

2. Preprocessing algorithm

To generate the structural information from the numerals, we use contour smoothing and linearization. For any component of the image, let the starting point be the upper left corner. Then by using Freeman 8-direction chain code based edge tracing algorithm, the outer contour of the k^{th} component of the image can be represented as:

$$C_k = \{c_0, c_1, \ldots c_n\} \quad (1)$$

Where c_i is the Freeman direction code relative to previous point and i is the index of the contour pixel. Then the difference code is defined as

$$d_i = c_{i+1} - c_i \quad (2)$$

In the smoothed contour $|d_i|$ equals 0 or 1 [5]. The set of smoothed contour can be converted to lines consisting of ordered pixels. If direction chain code of a smoothed contour is

$$\{c_l^{lin}[i]\ (i= 0,1,\ldots\ldots\ldots..(n_l^{lin}-1))\} \quad (3)$$

(where lin is the lin-th line of a smoothened contour and n is the number of points of the ln-th line.) then a linereazied line is defined with the following property

$$d_{ij}=c_l^{lin}[i]-c_l^{lin}[j]\ (i=0,..k-1),\ (j=0,\ldots\ldots,k-1), \quad (4)$$

then for a linearized line

$$|d_{ij}| \le 1\ (i=0,..k-1),\ (j=0,\ldots\ldots k-1), \quad (5)$$

In other word, a linereazied line contains only two elements whose chain codes meet Eq. 5.

Next, depending on the value of the direction codes of two consecutive lines the structural codes are assigned to the start or end points of the linereazied lines of the contours. The structural points describe the convex or concave change in different chain code direction along the contour. For example "^" represent the convex changes in the chain code 4 for the linereazied lines. They

can be therefore used to represent the morphological structures of a contour. The total number of possible structure feature points used is 16. They are shown in Fig. 2.

After detection of the structural feature points, the binary image is thinned. There are many excellent thinning algorithm reported in the literature but we have used the work proposed by Datta and Parui [6]. Here an iterative thinning algorithm based technique is used for the purpose. First, one black pixel is chosen and its eight neighbor are analyzed to check whether it is a critical (deleting whom the image will be broken) or end pixel. If so, it is kept unchanged. Otherwise, the pixel is converted to white. This process is repeated until no more black pixel can be converted to white. The resultant image is a thin version of the given image. The thin image is then analyzed to find the junction (a black pixel having more then 2 neighboring black pixels) and endpoints (a black pixel having only one neighboring black pixels) of the image. The Contour ('.',) the thinned image ('*'), the end points ('E'), the junction points ('J'), and the structural points (characters shown in Fig. 2.) are shown in Fig. 3 for numeral seven (৭).

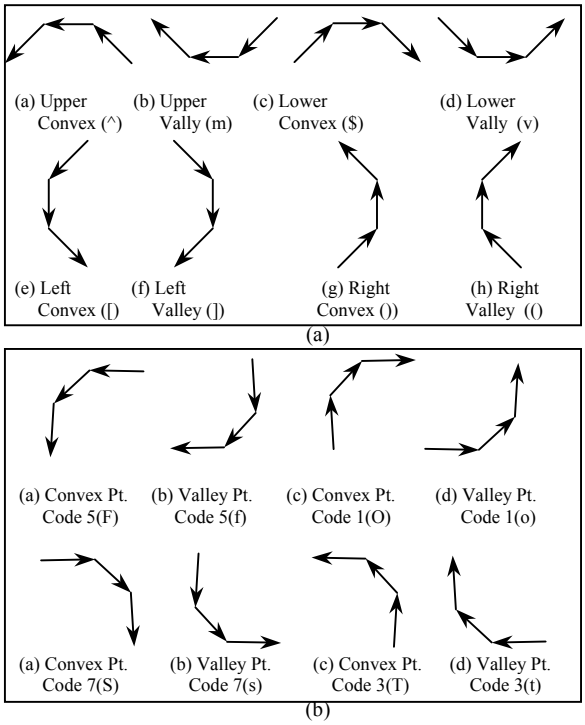

Fig. 2 The structural features and their codes are shown.

3. Reconstruction of broken handwritten numerals

The segments of a digit are divided into two categories, the main (the component having the largest area) and the adjacent component. The region where the main component lies is known as main region of a numeral. Before reconstruction of a broken digit, we have to select the end points to be joined. They are selected on basis of the following criteria:

1. Out of two end points in a selected broken point one should be in the main region and the other in the adjacent region.
2. The distance between the two selected end points should be less than 2.5 times the stroke width.
3. In between the pre-selected endpoints there should be some special structural points ("^", "v", "[" or ")").

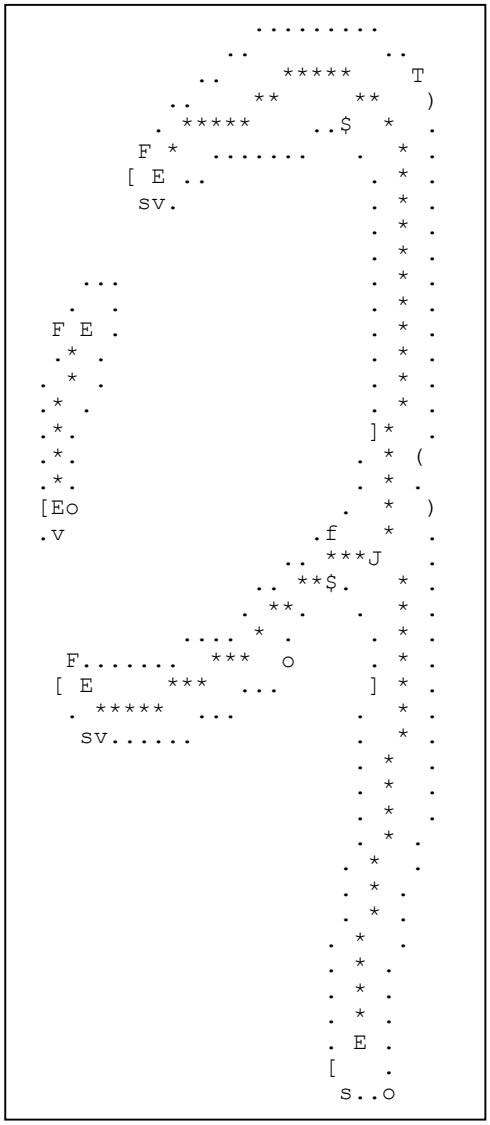

Fig. 3. The Contour ('.') the thinned image ('*'), the end points ('E'), the junction points ('J'), and the structural points (rest characters) are shown for numeral seven (৭) of Fig. 1.

After the end point pair is selected for joining, they are initially joined via a line. The width of the line drawn is the same as the stroke width of the character being joined.

Here, the initial line is drawn between the end points and then for every point on the line it is increased on both the sides to get the width of the line equal to the stroke width of the image. Some broken images and their reconstructed results are shown in Fig. 4.

4. Feature selection

The feature selection is a very important criterion in handwritten numeral recognition. They should be robust and easy to compute. Here we have used the features, which are mainly applied for joining the broken numerals. Some additional features such as reservoir feature [7] are used in fine-tuning of the initial coarse classification result.

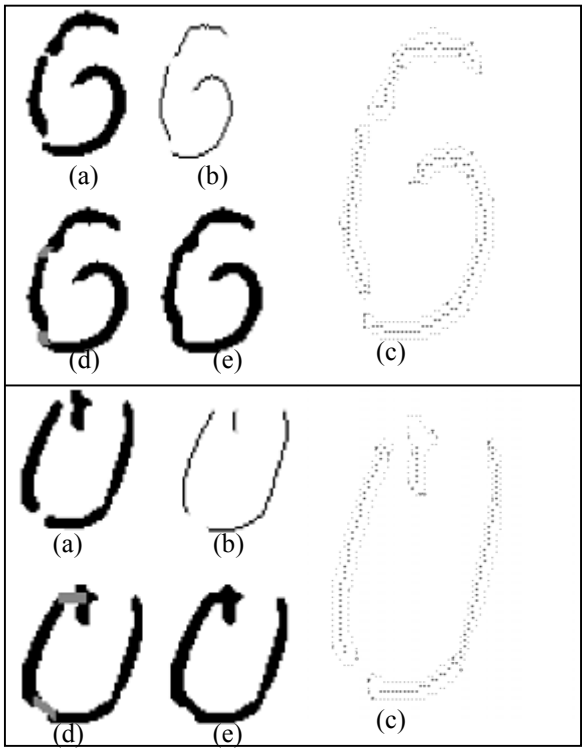

Fig. 4. Three sets of reconstructed numerals are shown. Here for each set (a) the original image, (b) the thinned version, (c) the contour with Structural points, (d) image after joining (added part is shown in gray scale), (e) the final binary image after joining of the broken part.

In the MLP base coarse classification we have used 21 normalized features. These features are based on:

1. 4 sets of (x, y) co-ordinates of the end points. For normalization of end points, the x and y coordinates are divided by length and width, respectively in such a way that the normalized value lies between 0.5 and 1 and if end points are less than 4, they are padded with 0 to fill the fields.
2. 3 sets of (x, y) co-ordinates of the junction points. They are normalized in the same way as end points.
3. The total number of end points and junction points. They are normalized in between 1 and 0.
4. (x, y) co-ordinates of the centre of gravity and signed Euclidean distance between centre of gravity and mid point of the bounding box of the image. Here x and y co-ordinates are normalized by dividing length and width, respectively. To normalize the distance between centre of gravity and mid point of the bounding box, it is divided by the diagonal, and if it lies in the upper half of the bounding box it is taken as positive, else negative.
5. Maximum length of the horizontal and vertical black run. Dividing them with length and width, respectively, normalizes them.

Other than the above features, some additional features are also used for by tree classifier based recognition. They are

1. Water reservoir based features.
2. Presence of structural points like "^", "$" etc. in the pre-selected regions of the image (see Fig. 2).
3. Presence of loops and their positions.
4. Number of components.

4.1 Water reservoir principle based features

The water reservoir principle is as follows. If water is imagined to be poured from a side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [7].

Top (bottom) reservoir: By top (bottom) reservoir of a component we mean the reservoirs obtained when water is poured from top (bottom) of the component. A bottom reservoir of a component is visualized as a top reservoir when water will be poured from top after rotating the component by 180°.

Left (right) reservoir: If water is poured from left (right) side of a component, the cavity regions of the component where water will be stored are considered as left (right) reservoir.

Here we will use the area (the area of the cavity region where water can be stored), the height (depth of water in the reservoir), the base line (line passing through the deepest point) of a reservoir, and flow direction (direction in which water overflows). For illustrations, see Fig. 5.

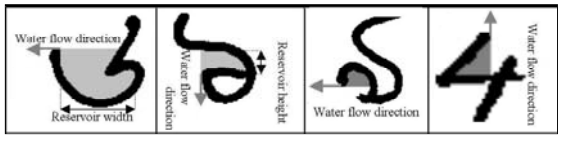

Fig. 5: Reservoirs obtained from top, left, bottom and right side of the components are shown.

5. Recognition procedure

5.1 MLP network

Based on the above normalized features, we use Multilayer Perceptron Neural Network based scheme for the recognition of Bangla numerals [8]. We now discuss different parts of the neural network used in the proposed scheme.

The Multi Layer Perceptron Network (MLP) is, in general, a layered feed-forward network, pictorially represented with a directed acyclic graph. Each node in the graph stands for an artificial neuron of the MLP, and the labels in each directed arc denote the strength of synaptic connection between two neurons and the direction of the signal flow in the MLP.

For pattern classification, the number of neurons in the input layer of an MLP is determined by the number of features selected for representing the relevant patterns in the feature space and output layer by the number of classes in which the input data belongs. The neurons in hidden and output layers compute the sigmoidal function on the sum of the products of input values and weight values of the corresponding connections to each neuron.

Training process of an MLP involves tuning the strengths of its synaptic connections so that it can respond appropriately to every input taken from the training set. The number of hidden layers and the number of neurons in a hidden layer required to design an MLP are also determined during its training. Training process incorporates learning ability in an MLP. Generalization ability of an MLP is tested by checking its responses to input patterns which do not belong to the training set.

Back propagation algorithm, which uses patterns of known classes to constitute the training set, represents a *supervised learning* method. After supplying each training pattern to the MLP, it computes the sum of the squared errors at the output layer and adjusts the weight values of the synaptic connections to minimize the error sum. Weight values are adjusted by propagating the error sum from the output layer to the input layer.

The present work selects a 2-layer perceptron for the handwritten numeral recognition. The number of neurons in input and output layers of the perceptron are set to 21 and 6, respectively since the number features is 21 and the number of possible classes in hand written numerals selected for the present case is 6. Here though the number of total character was 10 we have used only 6 classes in the output layer of the MLP. This is because the numerals zero (০) and five (৫) looks similar and these two numerals are considered as a single class. Similarly the four similar shaped numerals: one (১), three (৩), six (৬), and nine (৯) are considered as a single class. These similar shaped numerals are mainly confused/mis-recognized among themselves as can be seen from the confusion matrix shown in Table 2. The number of hidden units is 20, Back Propagation learning rate and acceleration factor is set to suitable values, based on trial runs. A network of 21-20-6 is thus finally designed.

5.2 Tree classifier

We in general recognize numerals based on their structural shape of the numerals in some particular regions. Based on these structural shapes we generate two tree classifiers for the recognition of the two multi-element classes obtained by NN. One class is formed by ০ and ৫, as shown in Fig. 6(a), and the other class is formed by ১, ৩, ৬, and ৯ as shown in Fig. 6 (b).

For the classification of ০and ৫ we find a reservoir and/or concave shape in the right side of the image. This shape is absent in zero (০) and present in five (৫) as shown in Fig. 6 (a).

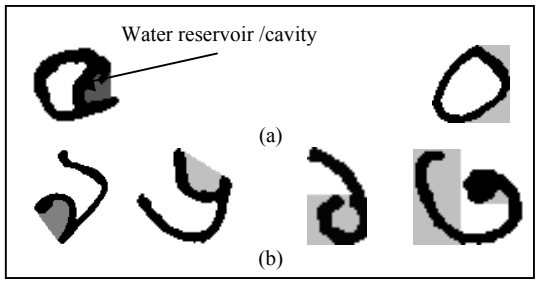

Fig. 6. Numerals recognized by tree classifier with their zone of interest are shown in gray shade. (a) for class zero (০), and five (৫) and (b) for class (১), three (৩), six (৬), and nine (৯).

In the other class we first find out the big cavity region on the lower side of the image by our water reservoir concept as shown in Fig. 7(a). At the base point of this reservoir the numeral is segmented to get the left and right part of the numeral as shown in Fig. 7(b).

After segmentation at the reservoir base of the numeral, we first consider the left part. If there is a particular concave-convex shape in the left side, it is nine (৯). Else, for the rest the members of the class we consider the right part. If there is a particular concave-convex shape in the right side it is six (৬). Now to recognize one or three we again consider the right part. If the height of the right part is greater than 60% that of its right part than it is three (৩), else if it is less than 30% that of its right part then it is recognized as one (১). If the height of the left part is between 30-60% of its right part than we again consider its right part for the final decision. If there exist a reservoir from bottom in right part then it is recognized as three (৩) else one (১).

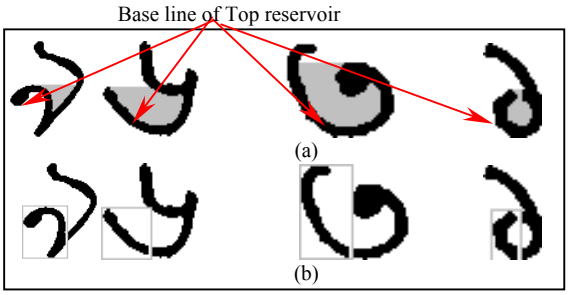

Fig. 7 (a) Numerals with reservoir are shown. (b) Numerals with their left part are shown.

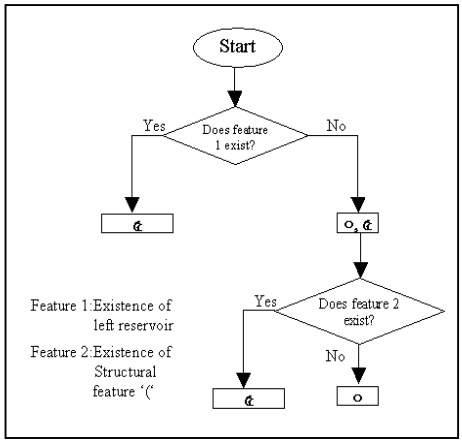

Fig. 8. Tree classifier for recognition of ০ and ৫ is shown.

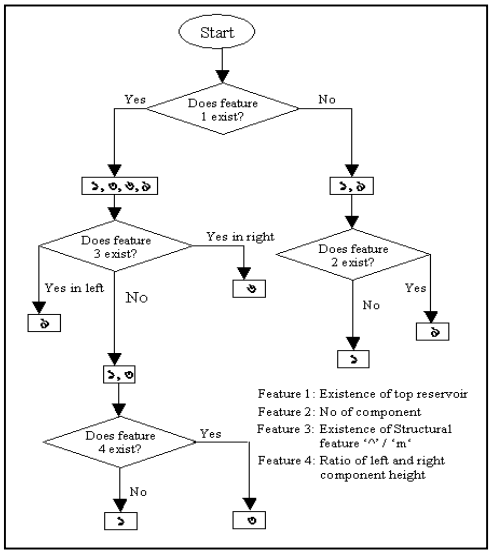

Fig. 9. Tree classifier for recognition of the class containing ১,৩,৬, and ৯.

6. Results and discussion

For the experiment of proposed numeral recognition approach, we considered gray-level image from Postal document collected from Cossipore post office of Kolkata, India. We also collected some data from individuals. The total size of the numeral database was 12410, and the number of the postal data was 2410. Among them, a dataset of 5000 (4250 from the individual writing database (IRD) and 750 from postal document) numerals were selected for training of the proposed recognition system and the remaining 7410 (5750 from IRD and 1660 from postal) numerals are used as test set.

From the experiment of broken numerals reconstruction module we note that our method can join broken line in 93.21% cases. In 2.59% cases it joined numerals wrongly, which were not actually broken. In 4.21% cases it failed to join the broken parts. The reason of failure was mainly due the distance (here we have used Euclidean distance and if it is less then 2.5 times of the stroke width they were joined) between the end points and absence of required end points for joining. Examples of some numerals where broken part are not joined by the proposed approach are shown in Fig. 10.

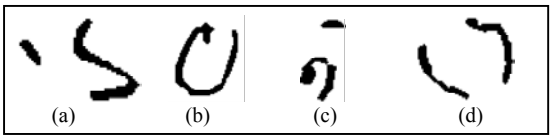

Fig. 10. Examples of numerals, where broken part is not joined. (a) six (৬), (b) zero (০), (c) nine (৯) (d) three (৩).

The binary image is first thinned and then its end points and junction points are detected. If it contains more then one end point then its outer contour is traced and the contour is smoothed and linereazied to find out the structural points. After finding the structural points, the numerals are checked for possible joining of the broken parts in the original image. If joining is done then the original image is thinned again to calculate the feature points for its initial recognition by the MLP.

The recognition result of the MLP obtained from the experiment on the above data set with all the numerals as separate class and grouping them in six class as stated above are shown in Table 1. The confusion matrix with initial 10-class on IRD dataset is shown in Table 2.

Table 1: Training accuracy, and testing accuracy on IRD and postal data are shown.

Data Set	Recognition rate		Rejection rate	
	10 class	6 class	10 class	6 class
Training	95. 45 %	99.52%	8.33 %	1.12%
IRD	91. 48%	97.19%	11.62%	3.79%
Postal	87. 51%	94.67%	14.29 %	5.93%

The results obtained from the tree classifier are given in table 3. Final results obtained after applying the tree classifier on the result of 6-class NN classifier are shown in Table 4. Also, the confusion matrix of the combined classifier is shown in Table 5.

Table 2: The confusion matrix obtained from MLP, with initial 10-class on IRD dataset.

	০	১	২	৩	৪	৫	৬	৭	৮	৯
০	426	2	0	5	0	**78**	1	0	0	0
১	0	346	3	**11**	2	0	2	0	3	8
২	0	4	450	0	0	0	0	0	0	5
৩	5	**13**	0	435	0	1	**20**	0	1	2
৪	9	3	0	0	459	7	0	5	2	3
৫	4	0	3	2	7	317	6	2	1	1
৬	1	1	1	**11**	0	0	430	0	0	**23**
৭	0	0	0	0	1	5	0	476	0	0
৮	0	0	5	1	1	3	1	1	470	0
৯	0	**30**	4	5	3	0	8	0	0	394

Table 3: Recognition result of the tree classifier.

Tree	Recognition rate	Rejection rate
Tree 1(০,৫)	98.673%	0.48%
Tree 2 (১,৩,৬,৯)	96.75%	2.32%

Table 4: Final recognitions result on Training, IRD and postal data.

Data Set	Recognition rate	Rejection rate
Training	97.13%	2.04%
IRD	95.07%	4.91%
Postal	92.17%	6.93%

Table 5: Confusion matrix of the Final result on IRD data.

	০	১	২	৩	৪	৫	৬	৭	৮	৯
০	472	2	0	3	8	5	0	0	0	1
১	4	416	3	8	2	1	14	0	1	10
২	1	3	439	1	1	1	3	1	5	5
৩	7	4	0	451	1	0	6	0	0	9
৪	2	1	4	0	448	10	1	0	1	1
৫	14	2	2	0	7	438	3	3	1	0
৬	1	5	0	13	0	2	461	0	2	5
৭	3	1	3	1	1	4	0	472	0	1
৮	0	0	0	0	1	0	3	0	**473**	0
৯	3	14	11	6	1	2	4	0	0	426

It can be noted from the result that there is still scope for improvement. Here the source of error is due to mis-recognition of the numerals by MLP and tree classifier. The reason for misclassification by the MLP is structural similarity between numerals and that of the tree classifier is because of the shape similarity. Some mis-recognized results are shown in Fig. 11. Here numeral six (shown in Fig. 11 (a)) is mis-recognized as three due to absence of the concave shape in the right side of the image where a convex shape should be in six in its ideal form. In Fig. 11 (b) the lower part could not be joined as it contains one segment only and this five is mis-recognized as zero. From the confusion matrix we see that the highest recognition accuracy is obtained for numeral eight (৮) (99.16%) and this is because it's different shape with the rest of the Bangla numerals.

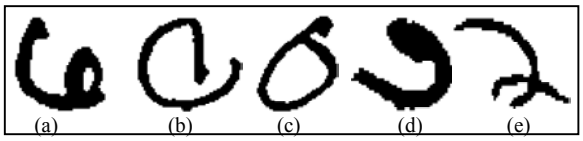

Fig. 11. Some mis-recognized numerals, (a) six (৬) is mis-recognized as three (৩), (b) five (৫) is mis-recognized as zero (০), (c) zero (০) is mis-recognized as five (৫), (d) three (৩) is mis-recognized as one (১), (e) one (১) is mis-recognized as nine (৯).

7. Conclusion

In this paper Bangla numeral recognition system is proposed for Indian postal automation. A combine classifier based on NN and tree classifier is proposed. We have tested our system on 12410 data and we obtained 94.21% overall accuracy from the proposed system.

Acknowledgement:

Partial financial support by Indo-French Centre for the Promotion of Advanced Research (IFCPAR) is greatly acknowledged.

References

[1] Bartnik D., V. Govindaraju, S. N. Srihari and B. Phan, "Reply Card Mail Processing", Proc. ICPR, Brisbane, Australia, pp. 633-636, 1998.

[2] Kim G., and V. Govindaraju, "Handwritten Phrase Recognition as Applied to Street Name Images", Pattern Recognition, 31, pp. 41-51, 1998.

[3] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: A comprehensive survey", IEEE Trans. on PAMI, Vol. 22, pp 62-84, 2000.

[4] K. Roy, U. Pal, and B.B. Chaudhuri, "Address Block Location and Pin Code Recognition for Indian Postal Automation", 2nd National Workshop on CVGIP, India, pp. 8-15, 2004.

[5] Donggang Yu and Hong Yan, "An efficient algorithm for smoothing linearization and detection of structural feature points of binary image contours", Pattern Recognition Vol. 30, pp. 57-69, 1997.

[6] A. Datta, and S. K. Parui, A Robust Parallel Thinning Algorithm for Binary Images", Pattern Recognition Vol. 27, pp. 1181-1192, 1994.

[7] U. Pal, A. Belaid and Ch. Choisy, "Water Reservoir Based Approach for Touching Numeral Segmentation," In Proc. Sixth ICDAR, pp 892-896, 2001.

[8] K. Roy et al., "An Application of the Multi Layer Perceptron for Handwritten Numeral Recognition" Proceedings of the Int. Conf. on CODEC, 2004, Kolkata, India