Recognition of Similar Shaped Handwritten Characters using Logistic Regression

Kinjal Basu
Indian Statistical Institute, Kolkata
Email: basukinjal@gmail.com

Radhika Nangia
Institute of Engineering and Management
Email: radhikanangia23@gmail.com

Umapada Pal
Indian Statistical Institute, Kolkata
Email: umapada@isical.ac.in

***Abstract*—Recognition of similar shaped characters is a difficult problem and in character recognition systems most of the errors occur in similar shaped characters. In this article we propose a generic method to differentiate between two similar shaped characters, which works well not only when the characters are rotated about its center, but also in the presence of noise. Rotation is taken care of by contour distance based approach and recognition is done based on logistic regression. We consider a training data set to estimate the parameters of the logistic model, and using these parameters we classify the test object. We have considered pairs of similar shape characters of Bengali script for testing our algorithm.**

***Keywords*-Handwritten Similar Character Recognition; Rotation Correction; Logistic Regression.**

I. Introduction

Recognition of characters has been a popular research area for many years because of its various application potentials. Different approaches have been proposed by the researchers towards Bengali character recognition and many recognition systems are available in the literature [7].

Although high accuracy is obtained from some of the systems, it may be noted that most of the errors are due to similar shaped characters. Recognition of these similar shaped characters is one of the difficult problems, and very few articles in literature deal with this issue [12], [13]. In this article we propose a technique to recognize similar shaped characters which works well in noisy situations as well as when the characters are rotated about a particular angle. The technique is based on logistic regression, a statistical regression technique that gives us the probability of belonging to a particular class, when a comparative study is performed.

Many methods of feature extraction are available in literature. Shadow code, fractal code, profiles, moment, template, structural (points, primitives), wavelet, directional feature, F-Ratio Weighted Feature [12], etc., are few of them. From the literature survey of the existing pieces of works on characters recognition, it has been noted that the weight (See Section 4) of each independent pixel has not been considered to enhance the recognition result. We introduce here such a technique based on logistic regression.

We initially start with the pre-processing of the data described in Section II, followed by the details of logistic regression in Section III. Explaining the two-category classification procedure in Section IV, we proceed to dealing with rotated images in Section V. Giving experimental results in Section VI, we conclude the paper in Section VII.

II. Pre-Processing of the Data

Since we are dealing with handwritten data, presence of a slanted or rotated image is an important factor we have to deal with. We try to correct the rotation present in the data using a contour distance based method described in section 5. Having the corrected image with us, we remove the excess white space and consider the minimum bounding box for the image. We further convert the different sized images into the same size (20×20 pixels). We have used several methods for this and then a comparative study was done based on several methods such as nearest neighbor interpolation, bilinear interpolation, bicubic interpolation [1] and Lanczos re-sampling [2] to find the optimum. We have seen that the classification is best for nearest neighbor interpolation. We then proceed to thinning using the graph-based thinning algorithm [11] because it suppresses shape distortion as well as false feature points. Lastly smoothing is done to suppress noise and small fluctuations. Here too, we have performed a comparative study on several filters to find the optimal one in our situation. The filters that we studied include - average, Gaussian, Laplacian, Laplacian of Gaussian and unsharp filter [4]. We have found that the Laplacian of Gaussian filter gives us the best results.

III. Logistic Regression

In statistics, logistic regression (sometimes called the logistic model or logit model) is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. It is a generalized linear model used for binomial regression.

After pre-processing of the images we have with us, a set of images belonging to each known category and a set of test images. Using the known images we estimate the regression parameters and then we use those parameters to calculate the probability of the test image to distinguish its class. We classify according to the maximum probability.

The Model and Set-up : The logistic function f is defined as

$$f(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

The input is z and the output is $f(z)$.

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.73

The variable z is usually defined as

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k$$

where β_0 is called the "intercept" and $\beta_1, \beta_2, \beta_3$, and so on, are called the "regression coefficients" of x_1, x_2, x_3 respectively. Logistic regression analyses binomially distributed data of the form

$$Y_i \ \sim B(n_i, p_i), \text{ for } i = 1, \ldots, m$$

where the numbers of Bernoulli trials n_i are known and the probabilities of success p_i are unknown.

The model proposes for each trial i, there is a set of explanatory variables that might affect the final probability. These explanatory variables can be thought of as being in a k-dimensional vector X_i and the model then takes the form

$$p_i = \mathrm{E}\left(\left. \frac{Y_i}{n_i} \right| X_i \right)$$

The logits, natural logs of the odds, of the unknown binomial probabilities are modelled as a linear function of the X_i.

$$\mathrm{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}$$

The unknown parameters β_j are usually estimated by maximum likelihood using iteratively re-weighted least squares.

IV. Two-Category Classification Procedure

In our set-up, we have a set of images belonging to each of the two classes. Each image after preprocessing is of the same size, for example, each image can be considered as a $m \times n$ matrix. Here we convert each of these matrices into a mn-vector. Thus we have the covariate matrix as

$$X = \left(\begin{bmatrix} X_1 & X_2 & \ldots & X_k \end{bmatrix}' \right)_{k \times mn}$$

where $k = n_1 + n_2$, n_i being the number of images belonging to the i-th class. Before we carry out the logistic regression, further cleansing of the data is done. These methods are described as follows:

A. Removal of Dependency

Note that we have the $X_{k \times mn} \quad (k > mn)$ matrix as described above. In certain situations, due to dependent variables we get $rank(X) < mn$. We remove these dependent variables by the following method.

We first convert the matrix X into its reduced row echelon form [9]. Note that the row reduced form has non zero elements only in the first r rows, where $r = rank(X)$. Also, for each of the non-zero rows, the position of the first 1 (while scanning from left), tells us that the variable at that position is independent. Thus noting those positions we get the list of variables which are independent.

For example consider the following 4×4 matrix

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \Longrightarrow A_{rref} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where A_{rref} denotes the reduced row echelon form of the matrix A. Note that $rank(A) = 2$. And correspondingly only the first 2 rows A_{rref} are non-zero. Furthermore, the first 1's occur in the 1st and 2nd column, in the two rows respectively. Thus out of the 4 variables, only these 2 are independent, and we remove variables 3 and 4.

B. Outlier Detection

Several methods of outlier detection have been discussed in literature [3], [10]. One of the easiest methods for detecting the outliers is through the Hat matrix H defined by $H = X\left(X'X\right)^{-1} X'$ which is a symmetric, idempotent matrix. An intuitive geometric argument reveals a link between the diagonal elements of H and the location of the points in the space. The set of p-dimensional points X_i that satisfy,

$$h_{X_i} = X_i \left(X'X\right)^{-1} X_i' \leqslant \max_i h_{ii} = h_0$$

determines an ellipsoid. This ellipsoid contains the smallest convex set enclosing the n observations. Consequently, we can say that the point X_i lies close to the bulk of the space formed by the explanatory variables if h_{X_i} is small. Since $trace(H) = p$, it is clear that the average value of h_{ii} is p/n. Using the cut-off given in Hoaglin and Welsch (1978) [5], we declare the i-th observation to be an outlier if $h_{ii} > 2p/n$.

C. Variable Selection

Observe that, even with a small image size of 20×20 pixels, we initially start with 400 variables, which, after removal of the dependent variables is still a pretty high number in almost all cases. Thus we employ a method of model selection in this situation, to reduce the number of variables, and only keep those which are highly significant in the regression set-up.

Note that there have been several papers in literature that deal with Model Selection. For our experiment, we have used the method of stepwise regression [6], considering the underlying model as logistic, to get a list of variables which are highly significant. We proceed with only these variables in the covariate matrix, say $X_{k \times s}$.

D. Classification using Logistic Regression

After cleansing the covariate matrix as described above, we estimate $\boldsymbol{\beta}$, the parameters of the logistic model (also considered as weights of the independent pixels), by $\hat{\boldsymbol{\beta}}$, as explained in Section III.

We test the unknown image by first converting it into the s-vector, say $\boldsymbol{x}$. (Keeping track of the variables which were removed during removal of dependency and model selection, and adjusting accordingly). Thus the mn-vector is accordingly reduced to the s-vector. Now we calculate the following probabilities

$$\text{Probability of belonging to } 1^{st} \text{ Class} = \frac{e^{\boldsymbol{x}'\hat{\boldsymbol{\beta}}}}{1+e^{\boldsymbol{x}'\hat{\boldsymbol{\beta}}}}$$

$$\text{Probability of belonging to } 2^{nd} \text{ Class} = \frac{1}{1+e^{\boldsymbol{x}'\hat{\boldsymbol{\beta}}}}$$

and classify according to the maximum.

V. Dealing with Rotated Images

The method of classification is based on the weights of independent pixels in the test image (described in Section IV). Thus if the image is rotated by a sharp angle in many cases we do get faulty results. Many handwritten characters are slanted or rotated by a particular angle. Thus in order to classify them properly we need to correct their rotation. In this section we describe a method to properly align the rotated characters using a contour distance based approach [8].

A. Contour Distance-Weight (CDW) Plot

When a character is rotated by different angles then the distance between the contour points and the Center of Gravity (CG) will not change, as CG does not change with the rotation of the charcter. Using this invariant characteristic we will try to find out the rotation that exists for each test character. This is discussed in details in later subsections. Now we describe how a contour-distance plot is calculated. Firstly, a point invariant of rotation is chosen as a reference point. Since CG is rotation invariant we consider CG as the reference point. The CG of the character (x_c, y_c) is calculated as

$$x_c = \frac{1}{n}\sum_{i=1}^{P} x_i \text{ and } y_c = \frac{1}{n}\sum_{i=1}^{P} y_i$$

where $(x_i, y_i), i = 1 \dots P$ are the P black pixels of the character. Now for each specific angle say θ we calculate the Euclidean distance of the last black pixel in that direction, say d_θ. We also calculate the number of times we crossed from black to white and from white to black in that direction (number of crossing points) and call it the weight w_θ of the angle θ . Note that its is important to consider this weight because in certain characters the difference lies within the outer contour for example য and ষ. Thus in order to differentiate such pairs this weight function must be attached. Thus for each angle θ our contour-distance weight is $d_\theta \times w_\theta$. As we vary θ from 0 to 2π, we get the whole plot. Details on how to calculate the outer contour points can be found in [8]. Figure 1 shows the CDW plot of the character ব

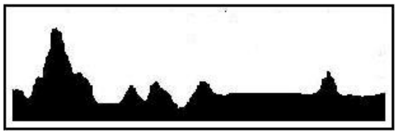
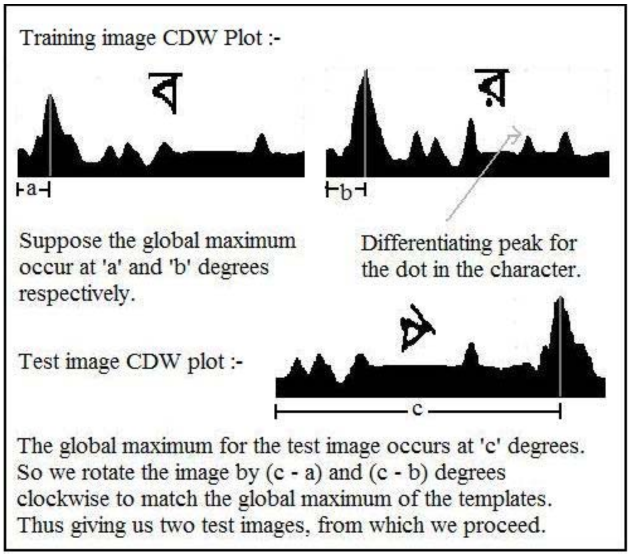


Figure 1. Example of a Contour Distance Weight Plot

B. Correcting for Rotation

Note that for every character we have with us the CDW plot as explained in the previous subsection. We use these CDW plots as reference. Now for the test image we similarly construct the CDW Plot. We know that this test image belongs to either of the similar group of test characters. So, we consider the position of the global maximum of the CDW plot of the test image and match it with the global maximum of the CDW plot of each of its similar classes. In this way we rotate the test image to generate different test images. Figure 2 shows this procedure in details for two characters ব and র where they differ only by a dot.



Suppose the global maximum occur at 'a' and 'b' degrees respectively.

Differentiating peak for the dot in the character.

The global maximum for the test image occurs at 'c' degrees. So we rotate the image by (c - a) and (c - b) degrees clockwise to match the global maximum of the templates. Thus giving us two test images, from which we proceed.

Figure 2. Matching of Global Maximum

We test these using the procedure as explained in the previous two sections. If both the results gives the same prediction we consider that to be our final outcome. If there is a mismatch in the prediction, we consider each such image and proceed as follows.

We rotate the test image through 5 degrees each starting from 5 to 2π. Thus we have 72 test images with us. For each image we calculate the test statistic $e^{x\beta}/(1+e^{x\beta})$, which is the probability of belonging to the first category. Suppose we denote the maximum and minimum by P_{max} and P_{min}, respectively. If $P_{max} > 1 - P_{min}$, i.e the maximum probability of belonging to the first class is more than the maximum probability of belonging to the second class, we classify it to the first class, otherwise to the second.

Note that we have used a 5 degree rotation on the images because as we are considering a 20 x 20 image size, considering center as (10,10) in order to get new pixel positions, we must rotate the image by

$$\tan^{-1}\frac{1}{20-10} = \tan^{-1}\frac{1}{10} \approx 5.7 \text{ degrees}$$

so truncating it to the nearest integer less than the value, such that we do not miss any information, we get the optimal turning angle as 5 degrees.

VI. Results

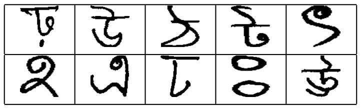
We considered 50 different characters (both handwritten and printed) of the Bengali script and calculated their pair-wise recognition rate. In each character group we had on an average 300 samples. We have used 75% of the data in the training dataset to help us estimate the parameter of the logistic model. We have then tested using the remaining 25% of the data. This method is repeated by selecting 1000 random subsets comprising of 75% of the data from the two classes being compared. The mean result is considered as the final detection rate.

Several results come to light from this extensive testing procedure. Both handwritten and printed dataset are considered separately for result computations. These are summarised below.

A. Results of Non-Similar Characters of handwritten data

When the characters were not similar to each other, in several pairs, we found that our algorithm gave 100% detection rate. Table I shows all the characters which were perfectly detected when first character of the pair is the consonant ক.

Table I
Example of characters which are perfectly recognized



B. Results of Similar Characters of handwritten data

The rate of detection of some similar shaped character pairs are given in Table II.

Note that in this situation we get the lowest value of 80.67%. But such situations are not out of the ordinary because even looking through the human eye it is sometimes difficult to classify. Thus due to presence of such cases the accuracy drops. Few examples of erroneous samples are shown in Table III.

Table II
Results of similar shaped characters

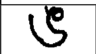
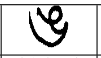
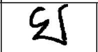
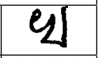
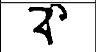
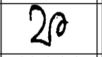
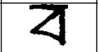
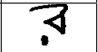
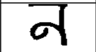
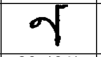
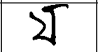
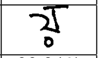
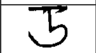
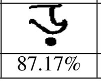
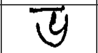
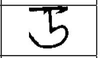
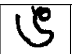
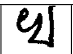
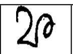
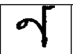
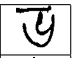
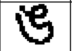
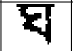
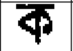
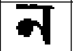
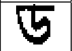
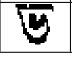
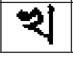
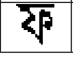
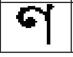
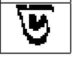
	Similar characters		Similar characters	
				
Detection Rate	80.67%	81.27%	90.86%	84.58%
				
Detection Rate	88.19%	88.85%	91.33%	92.20%
				
Detection Rate	82.96%	83.42%	89.10%	90.04%
				
Detection Rate	91.00%	87.17%	90.69%	84.83%

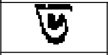
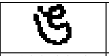
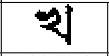
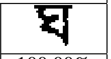
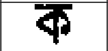
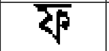
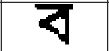
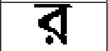
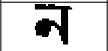
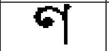
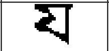
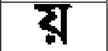
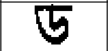
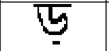
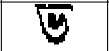
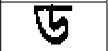
Table III
Examples of some erroneous samples

Actual Sample					
Recognised as					
Actual Class					

C. Results of Similar Characters of printed data

We have also applied this method on printed data sets. The result of similar shaped character pairs shown in Table II are given in Table IV. It can be noted that for the printed characters the results are very high (sometimes 100%) compared with handwritten similar shaped characters.

Table IV
Results of similar printed characters

	Similar characters		Similar characters	
				
Detection Rate	87.50%	100.00%	100.00%	100.00%
				
Detection Rate	100.00%	100.00%	100.00%	100.00%
				
Detection Rate	100.00%	100.00%	100.00%	100.00%
				
Detection Rate	95.00%	97.50%	100.00%	92.50%

D. Effect of Noise

In many situations we get the data with a high level of noise. We use the median filter to remove the noise present in the image. The results of printed and handwritten noisy characters are shown in Tables V and VI, respectively.

E. Effect of Rotation

The rotation algorithm had already been used in order to correct the slant or rotation in handwritten images. The results shown in Table II and IV are after the use of the

Table V
RESULTS OF SIMILAR PRINTED CHARACTERS WITH NOISE

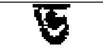
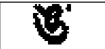
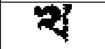
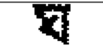
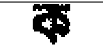
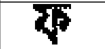
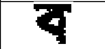
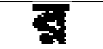
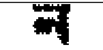
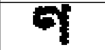
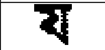
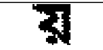
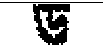
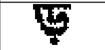
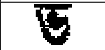
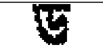
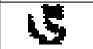
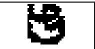
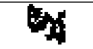
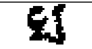
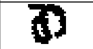
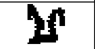
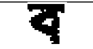
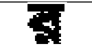
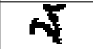
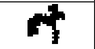
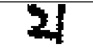
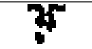
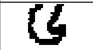
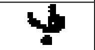
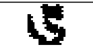
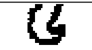
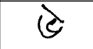
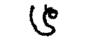
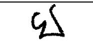
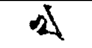
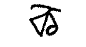
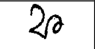
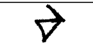
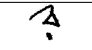
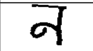
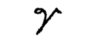
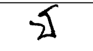
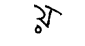
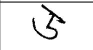
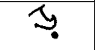
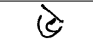
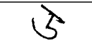
	Similar characters		Similar characters	
				
Detection Rate	82.50%	100.00%	87.51%	100.00%
				
Detection Rate	100.00%	100.00%	100.00%	100.00%
				
Detection Rate	100.00%	100.00%	87.50%	78.95%
				
Detection Rate	83.75%	88.51%	90.41%	85.20%

Table VI
RESULTS OF SIMILAR HANDWRITTEN CHARACTERS WITH NOISE

	Similar characters		Similar characters	
				
Detection Rate	76.37%	86.67%	79.31%	85.47%
				
Detection Rate	76.67%	78.50%	82.15%	79.30%
				
Detection Rate	75.31%	78.97%	76.67%	73.57%
				
Detection Rate	85.71%	82.14%	83.33%	78.57%

rotation algorithm. To show that it works well, we here artificially rotate the test images by a random angle from $-\pi/4$ to $\pi/4$. The results for handwritten characters are tabulated in Tables VII. Note that the accuracy drops a bit due to the sharp angles of rotation, however it is still comparable to the results shown in Table II.

Table VII
RESULTS OF ROTATED SIMILAR SHAPED HANDWRITTEN CHARACTERS

	Similar characters		Similar characters	
				
Detection Rate	80.54%	81.01%	90.41%	84.75%
				
Detection Rate	87.60%	88.00%	90.41%	92.30%
				
Detection Rate	82.45%	83.12%	89.00%	89.75%
				
Detection Rate	90.45%	86.70%	90.30%	84.75%

VII. CONCLUSION

In this article we have proposed and examined a method for similar shaped character recognition using logistic regression, which provides us with the probability of belonging to a particular class based on each of the independent pixels. It attaches a weight to each individual pixel's position which enables us to calculate the probability of the image to lie in a particular class. We have shown that the method works very well (giving perfect detection rate in many cases) for similar printed characters and relatively well for similar handwritten characters. Also, testing our method in the presence of noise, we get similar results in the case of printed characters, while it drops a bit considering the handwritten characters as expected. As for rotated data, we see that our method of rotation correction works well, since the detection rates are nearly the same, even when the handwritten images are rotated by a sharp angle.

We have mainly used this method for classifying similar shaped character pairs, but this method can also be generalized for the overall detection using a Multinomial logistic regression model instead of a Bernoulli model. We hope the proposed method will be useful for others in future research work especially when dealing with rotated images and noisy data.

REFERENCES

[1] W. Burger and M. J. Burge, Principles of digital image processing: core algorithms, Springer, 2009.

[2] C.E. Duchon, "Lanczos Filtering in One and Two Dimensions", Journal of Applied Meteorology, Vol. 18, No.8 , pp. 1016-1022, August 1979.

[3] F. E. Grubbs, "Procedures for detecting outlying observations in samples", Technometrics, Vol. 11, pp. 1-21, 1969.

[4] R. Haralick and L. Shapiro, Computer and Robot Vision, Vol. 1, Addison-Wesley Publishing Company, 1992.

[5] D. C. Hoaglin and R. E. Welsch, "The Hat Matrix in Regression and ANOVA", The American Statistician, Vol. 32, No. 1, pp. 17-22, 1978.

[6] R. R. Hocking, "The Analysis and Selection of Variables in Linear Regression", Biometrics, Vol. 32, pp. 1-49, 1976.

[7] U. Pal, R. Jayadevan and N. Sharma, "Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques", ACM Transactions on Asian Language Information Processing, in press.

[8] U. Pal and N. Tripathy, "A contour distance based approach for multi-oriented and multi-sized character recognition". Sadhana, Vol. 34, No. 5, pp. 755-765, 2009.

[9] A. R. Rao, and P. Bhimasankaram, Linear Algebra, 2nd Edition, Hindustan Book Agency, 2000.

[10] P. Rousseeuw, and A. Leroy, "Robust Regression and Outlier Detection", 3rd edition, John Wiley & Sons, 1996.

[11] S. Suzuki, N. Ueda, and J. Sklansky, "Graph-Based Thinning for Binary Images", IJPRAI, Vol. 7, No. 5, pp. 1009-1030, 1993.

[12] T. Wakabayashi, U. Pal, F. Kimura and Y. Miyake, "F-ratio Based Weighted Feature Extraction for Similar Shape Character Recognition", In Proc. 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 196-200, 2009.

[13] B. Xu, K. Huang, and C. Liu, "Similar Handwritten Chinese Character Recognition by Critical Region Selection Based on Average Symmetric Uncertainty", In Proc. 12th International Conference on Frontiers in Handwriting Recognition, pp. 527-532, 2010.