Recognition of Printed Kannada Numerals by Nearest Neighbor Method

Shantala Shettar
K.V.G College of Engg., Sullia
VTU University
Belgaum, India

Basavaprasad B.
Research and Development Centre
Bharathiar University
Coimbatore, India

Smt. Bhagya H. K
Associate Professor,
KVG College of Engineering
Sullia, Karnataka, India

Abstract- Numeral recognition is considered to be very prominent in most of the Character recognition researches. With respect to applications like number plate recognition and document processing the numerals are composed as a part of number plate images/application form type document images. This paper mainly focuses on eliminating language barriers that may arise while comprehending the regional language numerals by a non-regional user at the time of number plate recognition or other application form type document processing with special reference to Karnataka state. An algorithm is devised by incorporating the capabilities of functionalities of features the handwritten and printed Kannada numerals.

Keywords- Kannada numerals; OCR; numeral recognition; Normalization; Nearest Neighbor.

I. INTRODUCTION

Handwritten and printed Kannada numerals are integral part of most commonly used documents in real world. The recognition of printed and handwritten numeral recognition has been considered has an emerging research area spanned from Optical character recognition [1]. Many of the documents like application form for reservations, vehicle number plate images of various states, admission forms in Govt. Schools or colleges, historical documents pertaining to a state and all other types of documents may coexists the numerals with various other text. Especially kannada numerals can be observed in most of the real time documents to be used in various organizations for variety of their needs. Beginning with a simple literal like date, postal pin codes, identity card numbers, account numbers, register numbers, PAN card numbers, vehicle numbers, page numbers in a book or document and many other scenarios the recurrence of kannada numerals is high. Numerals are considered to be most standard uniform notation of numerals to be followed and understood by individuals belonging to different states/regions. Thus numerals are significant portions that reveal the accurate details of particular Place/individual very importantly based on a Place/individual’s ID card numbers, postal pin codes and vehicle number plate recognition. In a well civilized country like India there may be individuals belonging to varied states/ regions and working together in various Govt./Private Organizations without any regional differences but with some linguistic barriers. Individuals residing in rural areas are exposed more towards the regional languages, linguistic admirers and norms of certain state to use regional language numerals in some job areas also raised the need of recognition and conversion of the regional language numerals to eliminate the linguistic barriers among different state individual. This paper intends to devise algorithms for segmentation, classification and recognition of handwritten and printed numerals in application with features that provides the interface to recognize. Even though most of the research has been carried out in the area of handwritten/printed Kannada numeral recognition yet there is a need of experimenting it further to handle the various critics that may interrupt the faster and accurate recognition because recognition of a numeral plays a very typical role in identifying a particular individual. The current systems related to printed numerals have achieved almost 99% of accuracy where as in case of handwritten numerals it is still lagging below 90%. A generic numeral recognition system that can recognize both printed and handwritten Kannada numerals that can work with variety of datasets related to vehicle number plates or numerals that may occur in all types of general purpose documents etc. The handwritten and printed Kannada numerals that are used in various types of documents are different from the one that are used in current vehicle number plate recognition systems. Moreover, recognition of handwritten numerals is difficult because of the high Variability in writing styles of different Persons. The general purpose documents that are used in real time require a uniform conversion to one language. The numeral recognizer cum converter devised by us can be able to deal with all types of Kannada numerals that are used in different work environments. Especially, handwritten numeral recognition system that is used in postal pin code recognition requires this enhancement. All these various factors motivated us to develop this enhanced framework for Kannada printed and handwritten numeral recognition and translation system that can perform faster and accurate functioning.

II. LITERATURE REVIEW

There are numerous experimentations that are performed in the area of Kannada numeral recognition systems. Results of few of the experimentations are as discussed below. Dinesh Acharya et al. [1] has used ten segment string, water pool, horizontal or vertical strokes then end points as prospective features for Kannada handwritten numerals. U. Pal et al. [2] has proposed zoning method and directional chain code for Kannada numerals recognition. S.V. Rajashekararadhya et al. [3] have explained zone centroid and image centroid based angle feature extraction system for isolated Kannada numerals identification. Dhandra et al. [4] has proposed the features of pixel’s density for the detection of both handwritten and printed Kannada mixed numerals. Dhandra et al. [5] has explained spatial features and considered a feature vector of length 13 for handwritten utterly for Kannada numeral recognition. Rajput et al. [6] has proposed an approach for classifying Kannada, Telugu and Devanagari numerals using local and global structural features and probabilistic neural network classifier. Sivanandham et. al. [7] has discussed the various types of neural networks and its features. Dinesh Acharya et. al [8] has devised a method of recognizing isolated handwritten numerals using structural features and algorithm for classifying it. Anita pal et. al. [9] has proposed an algorithm to automatically recognize handwritten numerals using pixel density method that can reduce the classification and experimentation time. Rajashekarardhya et al. [10] has used the support vector machines and global features for the recognition of isolated hand written numerals. The literature survey, it is manifest that recognition of handwritten numerals is often done with respect to one type of application. Numerals recognition is a fascinating area of research; it is required to design a robust, accurate and faster handwritten and printed Kannada numeral recognition and translation system suitable for fulfilling varying needs in real time.

III. PROPOSED METHODOLOGY

The proposed system is intended to design a faster and accurate Kannada numeral recognition and using different features. The various steps involved in proposed numeral recognition system comprised of pre-processing, segmentation, feature extraction and classification, recognition and. Fig 1 depicts the details of steps involved in proposed system.

A. Pre-Processing

The pre-processing contains a set of operations that are performed on scanned input image document. A handwritten document must be scanned and converted into a suitable format for further processing. Pre-processing consist of different functions to clean the image and make it appropriate for carrying out the recognition process accurately. The different functions are,

i. Bounding box
ii. Thinning
iii. Noise removal
iv. Normalization

B. Segmentation

Images containing text are of great use in the real world. Extracting or modifying any text in the image requires the text to be segmented out from the document image. Hence Segmentation is an imperative and significant stage of any character recognition system. It separates the document image's text into lines, words and numerals which is an input for further stages of character recognition [19].

C. Thinning

Thinning is an important preprocessing step for many image analysis operations such as optical character recognition, fingerprint recognition and document processing. Thinning involves removing points or layers of an outline from a pattern until all lines and curves are a single pixel thick. The objective is to maintain the single pixel width along with perfect connectedness and topology. Different thinning algorithms involving different mathematical concepts and principles lead to different results. The general approach for extracting the skeleton or thinned image consists of removing all the pixels except those which belong to the skeleton. Despite a very intensive research in the last decades, development of thinning methods is still an active research area . The basic reason is probably the important role of thinning for improving the characteristics of the skeleton, thereby improving the effective algorithm performance.

```
Algorithm: Thinning
Input: Normalized image (numeral)
Output: Thinned image (numeral)
Method: Sequential contour (Pixel based) method
Step 1: Search the image (numeral) for horizontal line
Step 2: If found, thin the horizontal line by keeping middle line as it is and white wash all the rest of area else go to step 3.
Step 3: Search the black pixel in each and every row.
Step 4: If found,
(a) Compute the total number of columns before encountering the white pixel.
(b) Keep the center pixel black and rest all white then go to step 5,
Step 5: Check for search completion (40 rows and 30 columns).
Step 6: Stop.
```

D. Segmentation

Segmentation is a process that determines the constituents of an image. It is necessary to locate the regions of the document where data have been printed and distinguish them from figures and graphics. For instance, when reforming automatic mail sorting the address must be located and separated from other print on the envelope like stamps and company logos, prior to recognition. Applied to text, segmentation is the isolation of characters or words. The majority of optical character recognition algorithms segment the words into isolated characters, which are recognized individually.

```
Algorithm: Segmentation
1. Average character size of the current tackle is calculated, by scanning for segregated characters and noting their width and height.
```

```
2. If a column exists, check its pixel density. Else go to Step 10.
3. If the pixel density is zero, then segmentation point initiates. Evade ensuing columns with pixel density zero, until the beginning of the next character is encountered. Go to Step 2
4. If previous or next column's pixel density is bigger than current column, go to Step 5. Else return to Step 2
5. Calculate how many columns have been passed while last segmentation point.
6. If the number of columns is smaller than the usual size of the character, go to Step 2
7.If any later columns have a pixel density of zero, go to Step 2
8. Check if average pixel density of previous and ensuing columns is greater than that of the proposed point.
9. If Step 8 is true, then segmentation point originate. Reiterate by accessible to Step 2
10. End of segmentation procedure.
```

E. Noise Removal

The presence of undesirable dark pixels in the image gives rise to unwanted errors during character and numeral recognition. Noise is basically any unwanted interference that creeps into the image during the input stage or during the processing. For substantial accuracy this kind of interference has to be completely eliminated. There are a number of noise removal algorithms that are present in optical character recognition. The basic purpose of a noise removal algorithm is to first identify the noise and then remove it.

```
Algorithm: Noise Removal
Input: Thinned Image (numeral)
Output: Image (numeral) without noise
Method: Pixel based method
Step 1: Start
Step 2: Search the black pixel for each row and column (i.e., from row 1 to 40 and column 1 to 30), if found, check all the surroundings of  black pixel Otherwise go to step 4.
Step 3: If any surrounding pixel is black, convert that into a white pixel otherwise leave as it is.
Step 4: Make sure that search is complete for 40 rows and 30 columns or not.
Step 5: Stop.
```

F. Feature Extraction

After preprocessing, code the features (using MATLAB) and run the code for a number of image matrix inputs (samples of the numerals). In all, they have coded about 4 different kinds of features namely density feature, Left and right profile, Number of crossover points and Detection of horizontal and vertical line, so that we in effect have a m*n matrix, where n is the number of samples that we are giving and m is the number of features (in total they extracted 23 feature elements using the four basic features outlined above) as input for a particular numeral. Besides they have also assigned weights for different features. This process has been done for all the Devnagri Numerals (our numerals) and we have obtained a sample m*n (23*100) matrix. This is their database against which all the experimental input matrices are compared. Feature extraction is very typical and important in any numeral recognition research. The distinguished features of the numerals are mainly used by the classifier in order to identify and classify the numerals. The features of numerals are extracted and converted into a vector form, through which the neural network can recognize the appropriate numerals.

In this section, we have devised a feature extraction algorithm used for extracting the features from the segmented input data which leads towards efficient classification and recognition.

Algorithm: Features Extraction Algorithm

Input: Pre-Processed and segmented printed numeral Image.
Output: Feature matrix for Classification and recognition.

```
Step 1: Divide the image (numeral) into eight equal parts.
Step 2: Calculate total number of black pixels in each part.
Step 3: Store results in main feature matrix (8 features).
Step 4: Search for the black pixel from 1st, 11th, 21st and 31st rows, first from left to right side then right to left side and note the column  value where the black pixel is found during both the process.
Step 5: Store results in main feature matrix (4L + 4R = 8 features)
Step 6: Determine the maximum of all crossovers (i.e., white to black pixel and black to white pixel) between the rows: 1-10, 11-20, 21-& 31 - 40 and between the columns: 1-10, 11-20 & 21-30.
Step 7: Store results in main feature matrix (4R + 3C = 7 features).
```

G. Classification

Nearest Neighbor Classification: amid the numerous methods of overseen numerical pattern recognition, the Nearest Neighbor rule achieve reliably great presentation, deprived of a priori expectations about the circulations from which the training patterns are pinched. Therefore have used nearest neighbor method to classify the printed Kannada numerals. It involves a training set of both affirmative and undesirable cases. A fresh sample is categorized by computing the distance to the adjacent training case; the caveat of that point then determines the classification of the sample. The k-NN classifier extend this idea by delightful the k nearest points and turning over the notice of the majority. It is collective to select k small and odd to split ties (typically 1, 3 or 5). Greater k values help decrease the effects of noisy points inside the preparation data set, and the choice of k is often performed through cross-validation.

There are numerous procedures existing for the improvement of performance and rapidity of a nearest neighbor classification. One loom to this problem is to pre-sort the training sets in some way (such as kd-trees or Voronoi cells). Alternative clarification is to choose a subset of the training data such that classification by the 1-NN rule (using the subset) fairly accurate the Bayes classifier. This can lead to outcome in important speed enhancements as k can now be restricted to 1 and excess data points have been detached from the training set.

Algorithm: Nearest Neighbor
Input: Noise free image (Numeral)
Output: Identified numeral
Method: Nearest Neighbor
Step1: Start
Step 2: Calculate the Euclidian distance between each numeral of feature matrix with standard matrix.
Step 3: The numeral with the minimum distance is recognized as the actual number.
Step 4: Repeat the steps 2 and 3 for every numeral
Step 5: Stop.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The printed Kannada numerals are generated using Nudi 4.0 Kannada word processing software. For the implementation of proposed algorithm is coded using Matlab 7.0. Figure 4 shows a document image containing printed Kannada numerals which is the input. The proposed method generated the output as shown in the Figure. It can be noticed that for this particular font the proposed method could accurately recognize all the numerals.

The results and presented a technique for identification of printed Kannada numerals (digits) by density method. The pin code/zip code for mail addresses of Kannada script is very important. Though already a lot of has been done on this, tried best to implement a still better results. The work presented in this topic has very good potential application. The work of this method is font and size independent. In this topic have taken for 10 different fonts. Work can be extended for the remaining fonts.

NUMBERS		ERROR %	Cause of Error
1	1	0%	−
2	2	0%	−
3	3	0%	−
4	4	0%	−
5	5	0%	−
6	6	0%	−
7	7	0%	−
8	8	0%	−
9	9	0%	−
0	0	0%	−

V. CONCLUSION

During this work, have presented a technique for identification of printed Kannada numerals (digits) by density method. The pentode/zip code for mail addresses of Kannada script is very important. Though already a lot of has been done on this, tried best to implement a still better results. The work presented in this topic has very good potential application. The work of this method is font and size independent. In this topic have taken for 10 different fonts. Work can be extended for the remaining fonts.

The aim of the literature survey was to gain knowledge of the research already performed on the area of handwriting and printed numeral analysis. One approach which was thought to be adequate for the purpose, but which was not widely discussed in the literature was the use of statistical moment. This technique was used to compute the first set of feature vectors. It was discovered that no complete techniques were available for the task of determining similarities between handwritten numerals captures in an offline manner, and a new feature extraction method was hence created. Vector quantization has not been used in handwriting analysis, and was hence a novel approach.

REFERENCES

[1]. B. V. Dhandra, R. G. Benne, Mallikarjun Hangarge, “Kannada, Telugu and Devanagari Handwritten Numeral Recognition”.

[2]. A.K.Jain, Fundamentals of Digital Image Processing, ch. 7. Prentice-Hall, 1989.

[3]. B.Yu and A.K.Jain, “A robust and fast skew detection algorithm for generic documents,” in Pattern Recognition, pp. 1599-1629, 1996.

[4]. L. O’Gorman and R. Kasturi, “Document image analysis: An executive briefing,” 1999.

[5]. R.C. Gonzalez and R.E.Woods, Digital Image Processing, ch. 7, pp. 413-478. Prentice-Hall, 1994.

[6]. S. Haykins, Neural Networks A Comprehensive Foundation, ch. 4, pp. 156-175. Addison-Wesley, second ed., 2001.

[7]. U.Pal and B.B.Chaudhuri, “An improved document skew estimation technique,” in Pattern Recognition Letters, pp. 899-904, 1996.

[8]. Spitz, “Determination of the script and language content of document images,” Pattern Analysis and Machine Intelligence, pp. 235-245, 1997. 2Anil K. Jain and B. Yu, “Document representation and its application to page decomposition,” Pattern Analysis and Machine Intelligence, pp. 294-308, 1998.

[9]. B.B.Chaudhuri, U.Pal, and M. Mitra, “Automatic recognition of printed oriya script,” Sadhana, vol. 27, pp. 23-34, February 2002.

[10]. C. L. Tan, B. Yuani, and C.H. Ang, “Agent-based text extraction from pyramid images,” in International Conference on Advances in Pattern Recognition, Plymouth, UK, pp. 344-352, November 23-25, 1998.

[11]. D. Doermann, “The indexing and retrieval of document images: A survey,” Computer Vision and Image Understanding: CVIU, vol. 70, no. 3, pp. 287-298, 1998.

[12]. Cesarini, M. Gori, S. Marinai, and G. Soda, “Informys : A flexible invoice-like form reader system,” Pattern Analysis and Machine Intelligence, pp. 730-745, 1998.

[13]. Farrow, M. Ireton, and C. Xydeas, “Detecting the skew angle in document images,” SP:IC, vol. 6, pp. 101-114, May 1994.

[14]. G.S. Lehal and C. Singh, “A gurmukhi script recognition system,” in Proceedings of International Conference in Pattern Recognition, Barcelona, Spain, vol. 2, pp. 557-560, 2000.

[15]. Hochberg, P. Kelly, T. Thomas, and L. Kerns, “Automatic script identification from document images using cluster-based templates,” Pattern Analysis and Machine Intelligence, pp. 176-187, 1997.

[16]. L. O’Gorman, “The document spectrum for page layout analysis,” Pattern Analysis and Machine Intelligence, pp. 1162-1173, 1993.

[17]. M. D. Garris and D. L. Dimmick, “Form design for high accuracy optical character recognition,” Pattern Analysis and Machine Intelligence, pp. 653-656, 1996.

[18]. M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, “Syntactic segmentation and labeling of digitized pages from technical journals,” Pattern Analysis and Machine Intelligence, pp. 737-747, 1993.

[19]. R. G.Casey and E. Lecolinet, “A survey of methods and strategies in character segmentation,” Pattern Analysis and Machine Intelligence”, pp. 690-706, 1996.
[20]. Sadhana, vol. 27. Indian Academy Of Sciences, February 2002.
[21]. T. Wada, H. Ukida, and T. Matsuyama, “Shape from shading with interreflections under proximal light source - 3d shape reconstruction of unfolded book surface from a scanned image -,” in Proceedings of the Fifth International Conference on Computer Vision, 1995.
[22]. T.K.Ho, J.J.Hull, and S.N.Srihari, “Decision combination in multiple classifier systems,” Pattern Analysis and Machine Intelligence, vol. 16, no. 1, pp. 66-75, 1994.
[23]. T.V. Ashwin and P.S. Sastry, “A font and size-independent ocr system for printed kannada documents using support vector machines,” Sadhana, vol. 27, pp. 35-58, February 2002.
[24]. U. Pal and B. B. Chaudhuri, “Automatic separation of words in multi-lingual multi-script Indian document,” in Proceedings of International Conference on Document Analysis and Recognition (ICDAR), pp. 576-583, 1997.
[25]. Bansal and R.M.K.Sinha, “A devanagari ocr and a brief overview of ocr research for indian scripts,” in Proceedings of STRANS01,IIT Kanpur, 2001.
[26]. V. Bansal, “Integrating knowledge sources in devanagari text recognition,” doctoral thesis, IIT Kanpur, Department of Computer Science and Engineering, March 1999.
[27]. V. Wu and R.Manmatha, “Document image clean-up and binarization,” in Proceedings of SPIE conference one Document Recognition V, San Jose, California, January 24-30, 1998.