Abstract

Character segmentation has become a crucial task for character recognition in many OCR systems. It is an important step because incorrectly segmented characters are unlikely to be recognized correctly. For segmenting a cursive scripts leads more challenging because of presence of more touching characters. Kannada is one of the popular language in south India and also some of the letters in Kannada language are cursive in nature. In this paper, a new character segmentation algorithm for unconstrained handwritten Kannada scripts is presented. The proposed method is based on thinning, branch point and mixture models. The expectation-maximization (EM) algorithm is used to learn the mixture of Gaussians. We have used a cluster mean points to estimate the direction and branch point as reference points for segmenting characters. We have experimentally evaluated our proposed method on Kannada words and it has shown encouraging result.

keywords- Optical Character Recognition(OCR); Thinning; Branch Points; Mixture Models; Kannada script.

1. Introduction

Character segmentation is one of the most important decision processes for optical character recognition (OCR). Isolating individual alphabetic characters in the script image is often significant enough to make a decisive contribution towards the success rate of the overall system.

An OCR system may be designed to work for either of on-line and off-line purposes. On-line OCR systems collect input data by recording the order of strokes made by the write on an electronic bit-pad, and off-line OCR systems do the same by recording the pixel by pixel digital image of the entire writing with a digital scanner. OCR has a wide field of application covering handwritten document transcription, automatic mail address recognition, machine processing of bankchecks, faxes etc [1].

Off-line OCR of handwritten words has long been an active area research. In literature basic segmentation algorithms can be classified into three main categories: region based, contour based, and recognition- based methods [2]. Zhao et al. [12], proposed an improved algorithm for segmenting and recognizing connected handwritten characters. The method gradient descent mechanism is used to weight the distance measure in applying KNN for segmenting/recognizing connected characters in the left to right direction. Tan et al. [9], presented a new handwritten character segmentation method based on nonlinear clustering. segment the entire text line into strokes, the similarity matrix of which is computed according to stroke gravities. The nonlinear clustering methods are performed on this similarity matrix to obtain cluster labels for these strokes. According to the obtained cluster labels, the strokes are combined to form characters. Two-stage segmentation of unconstrained handwritten Chinese characters is reported in [11]. Maragoudakis et al [5], describes improved handwritten character segmentation by incorporating bayesian knowledge with support vector machines. Zheng et al. [13], presented character segmentation system based on C design and implementation. Sari et al. [7], presented off-line handwritten Arabic character segmentation algorithm based on morphological rules.

Basu et al [1], presented segmentation of off-line handwritten Bengali script. Pal et al [6], proposed touching numeral segmentation using water reservoir concept. A reservoir is a metaphor to illustrate the region where numerals touch. Reservoir is obtained by considering accumulation of water poured from the top or from the bottom of the numerals. At first, considering reservoir location and size, touching position (top, middle or bottom) is decided. Next, analyzing the reservoir boundary, touching position and topological features of the touching pattern, the best cutting point is determined. Finally, combined with morphological structural features the cutting path for segmentation is generated. Sharma and Singh [8], proposed segmentation of handwritten text in Gurumukhi script. The segmentation of half characters in handwritten Hindi text is presented in [4].

From the above literature survey many methods on handwritten character segmentation have been reported in English, Chinese and Arabic. Also, some works carried in Indian languages like Bengali, Gurumukhi. At the best of our knowledge it is of first kind in the literature for the handwritten Kannada character segmentation. Hence in this
study we proposed a new character segmentation algorithm for Kannada script.

The outline of the paper is follows: In section 2, we explain the properties of Kannada script. In section 3, explain the proposed methodology. Experiment results is presented in section 4. Finally conclusion are drawn at the end.

2. Properties of Kannada Script

Kannada script is written horizontally from left to right and an absent of lower and upper case like in English language. Moreover the Kannada characters are formed by combination of basic symbols, segmentation of the Kannada character is complex and challenging task & increased character set, it contains Vowels, Consonants Compound characters. Some of the character may overlap together. Kannada text is difficult when compared with Latin based languages because of its structured complexity. Moreover, Kannada language uses 49 phonemic letters, shown in Figure 1 it is divided into 3-groups, Vowels (Swaragalu- Anusvara (o), & Visarga (:)), Consonants (Vyanjanagalalu-34) and modifier glyphs (Half-letter) from the 15 vowels are used, to alter the 34 base consonants, creating a total of (34*15)+34=544 characters, sample of modifier glyphs is as shown in Figure 2, additionally a consonants emphasis glyph called Consonant conjuncts in Kannada (vattakshara), exists for each of the 34 consonants. This gives total of (544*34)+15=18511 distinct characters [10], samples of extra modifiers shown in the Figure 3.

3. Proposed Method

This section presents the proposed methodology of unconstrained handwritten Kannada character segmentation. Initially, all components in a word image are detected by connected component analysis (CCA) algorithm which is as shown in the Figure 4. For a component $c_i$, its height and width of a component are represented by $h_i$ and $w_i$ respectively. From this process the components those having average and below average height and width components are segmented, the remaining components are considered as touching components. To segment these touching components we follow three steps namely: Thinning, Branch Points and Mixture Models. All these steps are explained in following subsections.

3.1. Thinning and Branch Points

The characteristic of Kannada script is cursive in nature and also from the statistical analysis most of the touching portion can be find within the half height from the bottom of character. These touching causes mainly because of modifiers and extra modifiers. To find this touching portion in the components we have applied morphological thinning operation to for further process, which is as shown in the

Figure 1: Kannada language 49 phonemic letters.

Figure 2: sample of Kannada modifier glyphs (Diacritics).

Figure 3: Consonant conjuncts in Kannada (vattakshara).

Figure 4: (a)Input image (b) Result obtained after applying CCA algorithms.

Figure 3: Consonant conjuncts in Kannada (vattakshara).
Figure 5. Then, the thinned image is used to find the touching portion using branch points from templates of branch point identifiers. In this work we have used 16 different branches of templates and each one has four cross points, Figure 6 shows the several occurrences of 16 types. Figure 7 shows the branch points present in the thinned image. Selection of best branch point from many point for accurate segmentation is the main task. For this purpose, we analyze statically most of the touching portions by recognizing the right half of the width of component. Hence, we choose right most branch point is the segmentation point of the components. After getting the segmentation point, it is not straightforward way to cut touching portion like horizontal or vertical direction. This is due to the circular nature of Kannada script. To resolve this issue, we have used Mixture Models to find the angle of direction to accurately segment the touching characters.

3.2. Mixture Models

Mixture models use a set of data points as a input for clustering input data. To find a clusters in a set of data points is a considerable problem. To obtain marginalization from joint distribution over observed and latent variables is relatively complex. To solve this problem use of latent variable in a mixture distributions in which the discrete latent variables can be interpreted as defining assignment of data points to specific components of the mixture. EM algorithm is a one of the technique for finding maximum likelihood estimation in latent variable [3].

Let we have dataset of observation $x_1, x_2, ..., x_n$. represent this dataset as an $n \times d$ matrix $X$ in which the nth row is given by $X^n_T$, similarly the corresponding latent variables will be denoted by an $n \times k$ matrix $z$ with rows $z^n_T$. The Gaussian mixture of conditional probability of $z$ is given by

$$
\gamma(z_k) \equiv p(z_k = 1|x) = \frac{\pi_k \eta(x|\mu_k, \sum_k)}{\sum_j \pi_j \eta(x|\mu_j, \sum_j)}
$$

(1)

where $\pi_k$ and $\gamma(z_k)$ is the prior probability of $z_k = 1$ and the quantity of corresponding posterior probability of observed $x$ respectively, $\mu_k$ is a mean point associated with $k^{th}$ clusters.

The log likelihood function of i.i.d(independent and identically distributed) for data points is given by

$$
\ln p(x|\pi, \mu, \sum) = \sum_{n=1}^{N} \sum_{k=1}^{K} \ln \eta(x_n|\mu_k, \sum_k).
$$

(2)

To obtain Maximum-likelihood function setting derivatives of $\ln p(x|\pi, \mu, \sum)$ w.r.t means $\mu_k$ of the Gaussian components to zero.
\[ 0 = - \sum_{n=1}^{N} \frac{\pi_k \eta(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{k} \pi_j \eta(x|\mu_j, \Sigma_j)} \sum_k (x_n - \mu_k). \quad (3) \]

The posterior probability, or responsibilities, given by (1) appear naturally on the right-hand side

\[ \gamma(z_{nk}) = \frac{\pi_k \eta(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{k} \pi_j \eta(x|\mu_j, \Sigma_j)} \]

Multiplying by \( \sum_{k} \) and rearranging we obtain

\[ \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \quad (5) \]

where \( N_k \) is defined as the effective number of points assigned to cluster \( k \).

\[ N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \quad (6) \]

The mean \( \mu_k \) for the \( k^{th} \) Gaussian component is obtained by taking a weighted mean of all the points in the data set, in which the weighting factor for data point \( x_n \) is given by the posterior probability \( \gamma(z_{nk}) \) that component \( k \) is responsible for generating \( x_n \).

To obtain a maximum likelihood solution for covariance matrix of single Gaussian is given by

\[ \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T \quad (7) \]

The mixing coefficient for the \( k^{th} \) component is obtained by an average responsibility which that component takes for explaining the data points.

\[ \pi_k = \frac{N_k}{N} \quad (8) \]

General procedure for EM algorithm

1) Initialize the parameters \( \mu_k, \Sigma_k \) and \( \pi_k \) and evaluate the initial value of the log likelihood.
2) **E** Step Evaluate the responsibilities using Eq.(1) with current parameter values.
3) **M** Step Re-estimate the parameters using the current responsibilities:

\[ \mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \]

\[ \Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \]

\[ \pi_k^{new} = \frac{N_k}{N} \]

4) Evaluate the log likelihood:

\[ \ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \eta(x_n|\mu_k, \Sigma_k) \quad (9) \]

and check for convergence of log likelihood. If the convergence criterion is not satisfied iterate from step 2.

Means of \( k \) clusters \( \mu_k, \forall k = 1, \ldots, K \), is then used for estimating the skew of a segmented touching components. Selecting \( k \) value is a highly subjective in nature. Therefore, we fix the value of \( k \) as 2. Figure 8 depicts the mean points obtained for the input skewed word using mixture-of-gaussians. In this work we have used skew estimators as Linear Regression Analysis(LRA) to estimate the skew angle of the word. Finally, with the reference to skew angle direction and branch point we segment the touching characters which is as shown in the Figure 9.

![Figure 8: Result obtained after EM algorithm.](image1)

![Figure 9: Estimate the angle of direction: (a) angle of direction showed in line (b) segmented touching character based on the estimated angle.](image2)
4. Experimental Results

This section presents the results of the experiments conducted to study the performance of the proposed method. The method has been implemented in MATLAB 10.0 on a Core2duo processor with 1GB RAM. For our experiment we have collected a data set comprising 400 handwritten Kannada words. Most of the words are presented with one or two touching components. Figure 10 shows the samples of word images presented in dataset. To find the segmentation accuracy, we adopted a segmentation-base strategy, in that explicit segmentation is used to segment a word into an ideal parts of isolated characters. The obtained segmentation accuracy of the proposed method is 85.5%. The successful segmentation results are shown in Figure 11. The proposed method is not so robust to challenges like, if a branch point is encountered which is not present in the touching portion of the component, which is shown in Figure 12 and for more than two components touching each other, which is shown in Figure 13.

Figure 10: Sample images of Kannada words.

Figure 11: Successful result obtained for proposed method.

Figure 12: Failure case of the proposed method: branch points are not present in the touching portion of the component.

Figure 13: Failure case of the proposed method: more than two components touching each other.

5. Conclusion

In this paper, a novel Kannada character segmentation algorithm is presented. The proposed method is based on the thinning, branch points and mixture models. The thinning is the most commonly adopted technique to skeletonize an input image and is used to find the branch points present in an image. From the reference of the best branch point and skew angle obtained from the mixture models, we have segment the characters from touching characters. The proposed method is tested on handwritten Kannada words and has shown encouraging results. In future we plan to work with more then two components are overlapping words.

References


