Offline Handwritten Kannada Text Recognition by Integrating Multiple Contexts

Sandhya C Sannamani
Dept. of Computer science & engineering.
Mangalore Institute of Technology & Engineering, Mangalore, India

Bharath Kumar M
Dept. of Computer science & engineering.
Mangalore Institute of Technology & Engineering, Mangalore, India

Nagesh H R
Dept. of Computer science & engineering.
Mangalore Institute of Technology and Engineering, Mangalore, India

ABSTRACT
Handwritten character recognition has received extensive attention in academic and production fields. The recognition system can be either online or off-line. There is a large demand for handwritten text recognition and hand written documents. This paper describes an effective approach for the offline recognition of handwritten Kannada texts. Under the general integrated segmentation-and-recognition benchmark with character over-segmentation and Recognizes the handwritten Kannada text, whenever there is multiple contexts in the text pattern and it is independent of size, slant, orientation, and translation, this approach investigates three important issues: candidate path evaluation, path search, and parameter estimation.
In the path evaluation method, we integrate multiple contexts of the text (character recognition scores, geometric and linguistic contexts) from the Bayesian decision view, and convert the classifier outputs to posterior probabilities using confidence transformation. In path search, refined beam search algorithm is used to improve the search efficiency and use a candidate character augmentation mechanism to improve the recognition accuracy. The combining weights of the path evaluation function are optimized by supervised learning using a Maximum Character Accuracy criterion. This method evaluates the recognition performance on Kannada handwritten text images, which contains Kannada letters, words and sentences. The experimented result shows that confidence transformation and combining multiple contexts improve the text line recognition performance, efficiency and throughput significantly.

Keywords
Handwritten Kannada text recognition, Bayesian decision view, Kannada letters and words, refined beam search algorithm, over segmentation, candidate character augmentation.

1. INTRODUCTION
Handwritten Kannada character recognition (HKCR) has received extensive attention in academic and production fields. HKCR is the important area in image processing and pattern recognition. The recognition system can be either on-line or off-line. In on-line handwriting recognition words are generally written on a pressure sensitive surface (digital tablet PCs) from which real time information, such as the order of the stroke made by the writer is obtained and preserved. This is significantly different to off-line handwriting recognition where no dynamic information is available. Off-line handwriting recognition is the process of finding letters and words are present in digital image of handwritten text. It is the subfield of optical character recognition (OCR).

Handwritten Kannada text recognition (HKTR) is a challenging problem due to the slant and orientation of character in the text, the diversity of writing styles, the character segmentation difficulty, and the unconstrained language domain. Fig.1 shows an example of a Kannada handwritten page. The different styles of handwritten Kannada characters bring difficulties to efficient and effective recognition. The divergence of writing styles among different writers and in different geographic areas aggravates the confusion between different letters. Handwritten Kannada text recognition is particularly difficult because the characters cannot be reliably segmented prior to character recognition. The difficulties of character segmentation originate from the variability of character size and position, character touching and overlapping. A text line of Kannada handwriting must be recognized as a whole because it cannot be trivially segmented into words (there is no more extra space between words than between characters). Sometimes Handwritten text recognition is more difficult than printed text recognition, bank check recognition and mail address reading because the lexical constraint is very weak: Under grammatical and semantic constraints, the number of sentence is infinite.

Due to the number of character touching styles, number font sizes, diversity of slants and the infinite sentence of Kannada texts, HKTR can only be solved by segmentation-based approaches using character models [24], preferably by explicit segmentation, also called over segmentation, which can take advantage of the character shape and overlapping and touching characteristics to better separate the characters at their boundaries. The result of over-segmentation is a sequence of primitive segments, each corresponding to a character or a part of a character, such that candidate characters can be generated by concatenating consecutive segments. The candidate character sequences [25] can be represented in a network called a candidate lattice [14], and each candidate segmentation path in the lattice can be split into many segmentation recognition paths by assigning character classes to the candidate characters. The result of character segmentation and recognition is obtained by evaluating the paths in the lattice and searching for the optimal path. In integrated segmentation-and-recognition, the candidate segmentation-recognition paths are usually evaluated by combining the character recognition scores (classifier outputs), geometric context, and linguistic context [16].

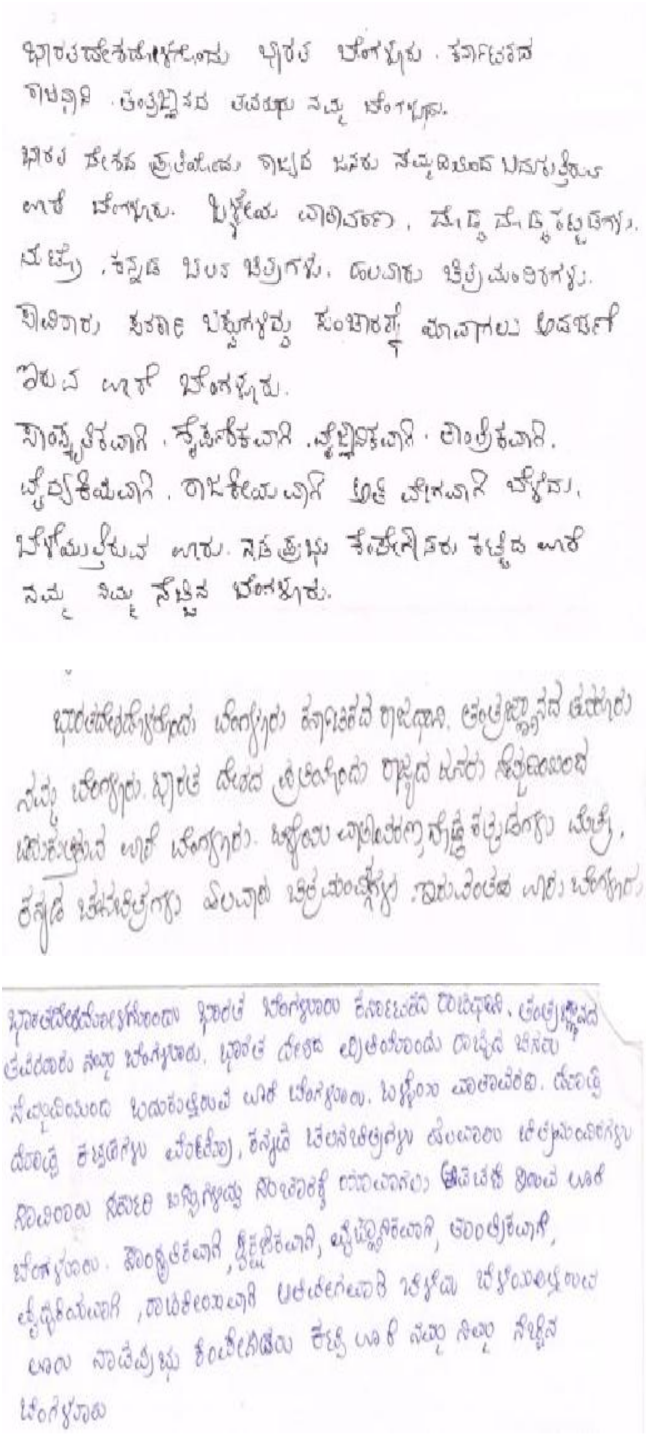


Fig. 1: A pages of scanned handwritten Kannada text.

Many mechanisms have been made this direction, but there has not been a complete satisfactory solution. The existing methods either integrated incomplete contexts or combined the contexts heuristically without optimizing the combining weights [23], [24]. Zhou et al. optimize the combining weights using the conditional random field (CRF) model [14], which is hard to incorporate into language models of higher order than the bi-gram. Zhu et al. optimize the combining weights using the genetic algorithm (GA) [15], which is computationally expensive and is sensitive to some artificial parameters. The previous works have addressed handwritten text (character string) recognition from different viewpoints and have contributed various techniques. However, none has investigated these techniques comprehensively and integrated them in a high-performance system for other languages handwritten text recognition. In this study, we investigate three key issues of integrated segmentation-and-recognition for HKTR: candidate path evaluation, path search, and parameter estimation [25]. By elaborating the techniques for these issues, we achieved significant improvements on unconstrained handwritten Kannada texts. In path evaluation, we integrate character recognition scores, geometric context, and linguistic context from the Bayesian decision view [25], and convert the classifier outputs to posterior probabilities via confidence transformation (CT). In path search, a refined beam search algorithm is used to improve the search efficiency and, meanwhile, a candidate character augmentation (CCA) strategy is applied to benefit the recognition accuracy. To balance the multiple contexts in path evaluation function, we optimize the combining weights on a data set of training text lines using a Maximum Character Accuracy (MCA) criterion. We evaluated the recognition performance on unconstrained Kannada handwritten characters and texts, and demonstrated superior performance by the proposed methods.

2. RELATED WORK

In the context of handwritten text or character recognition, many techniques are used to the related concepts of over-segmentation, character classification, confidence transformation, language model, geometric model, path evaluation and search, and parameter estimation. For over-segmentation, connected component analysis has been widely adopted, but the splitting of connected (touching) characters, diversity of character styles, orientation, slant has been a main concern [15], [16]. After generating candidate character patterns by combining consecutive primitive character segments, each candidate pattern is classified using a classifier to assign similarity/dissimilarity scores or issues to some character patterns. Character classification involves character normalization, feature extraction, and classifier design. The state-of-the-art methods have been reviewed in [17]. For classification of Kannada characters with a number of letters, words and sentences, the most popularly used classifiers are the modified quadratic discriminant function (MQDF) [18] and the nearest prototype classifier (NPC) [19]. The MQDF provides higher accuracy than the NPC but suffers from high expenses of storage and computation. Transforming the similarity/dissimilarity measures output by classifiers to probabilistic confidence measures can benefit from fusing multiple classifiers or fusing multiple patterns, as has been demonstrated in previous works. In character string recognition, Jiang et al. [20] transformed classifier outputs to confidence values under the soft-max framework. Li et al. [21] used the logistic regression model for confidence transformation.

Language models are widely used in speech recognition, machine translation, handwriting recognition, and so on [12]. The most popular language model is the n-gram, which characterizes the statistical dependency between characters or words. Character-level n-gram models have been popularly used in character string recognition (e.g., [23], [24]) Word-level and hybrid language models were used in post processing for correcting recognition errors after character segmentation, but have been rarely used in integrated segmentation-and recognition [25].

In addition to the character recognition scores and linguistic context, the geometric context also plays an important role in character string recognition, particularly for disambiguating character segmentation [23],[24] Zhou et al. elaborated the geometric context models into unary and binary, character class dependent and class-independent models in online handwriting Recognition [24], the geometric context models for offline handwriting and applied to transcript mapping of handwritten Kannada documents. Various methods have been used for HKTR process, whereas each approach provides solution only for few character sets. Challenges still prevails in the recognition of normal as well as abnormal writing, slanting characters, similar shaped characters, joined characters, curves and so on during recognition process.

A key issue in character string recognition is to design an objective function evaluating each candidate segmentation recognition path. The path evaluation function is hoped to be insensitive to the path length (number of characters on the path). The summation of classifier output similarity/dissimilarity scores or product of class probabilities is not appropriate since this is biased to short paths. Normalizing the summation or product by the path length overcomes the bias problem, but this normalized form does not enable optimal path search by dynamic programming (DP). Beam search can be used instead, but does not guarantee optimality.

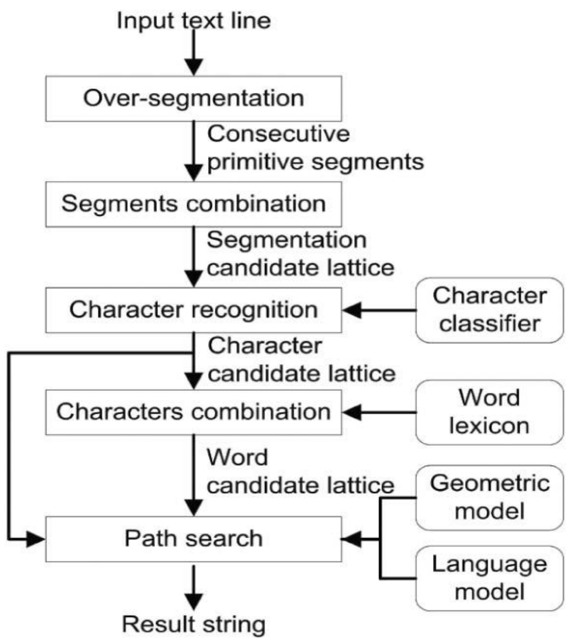


Fig.2: System diagram of handwritten Kannada text line recognition.

Another way to overcome the path length bias is to add a compensative constant in the summated path evaluation function, but the constant needs to be estimated empirically. Wuthrich et al. [8] called this constant a word insertion penalty, and Quiniou et al. [12] also used this constant to control the deletion and insertion of words. Another effective way is to weight the character classification score with the number of primitive segments forming the character pattern [25], [15], motivated by the variable duration HMM of Chen et al. [9]. This not only makes the number of summated terms in the path evaluation function equal the number of primitive segments (and thus independent of the path length), but also preserves the summation form and enables optimal path search by DP.

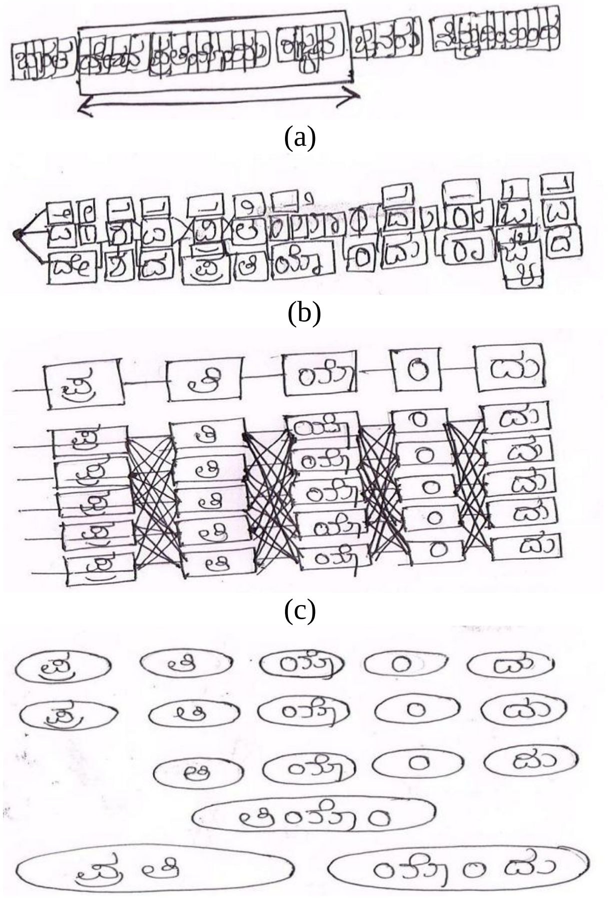
In weighted combination of context models for path evaluation, the weights were sometimes determined by trial and error. Some works have applied the supervised learning approach to estimate the weights by optimizing a string recognition criterion. The search of optimal path in Kannada character string recognition is not trivial because of the large number of candidate segmentation-recognition paths. The search is further complicated when using word-level language models because the word segmentation is again a combinatorial problem [25]. The speech recognition community has contributed many efficient search algorithms based on dynamic programming and some variants (e.g., beam search). The beam search strategy provides a good tradeoff between efficiency and accuracy. The character-synchronous beam search strategy is appropriate for lexicon-driven string recognition, while the frame-synchronous (also called as time-synchronous in speech recognition) strategy is appropriate for lexicon-free string recognition.

In character string recognition, the pruning or augmentation of character classes affects the search efficiency and accuracy. Ideally, a candidate character pattern is assigned as few classes as possible by the classifier, including the true class. For Kannada handwriting, it often entails a large number (e.g., several hundred) of candidate classes to guarantee a high probability of including the true class, however. This complicates the search space on one hand. And on the other hand, may deteriorate the recognition accuracy because there are too many wrong classes competing with the true class. Therefore, some works have attempted to reduce the candidate classes output by the classifier by confidence evaluation, and some other works attempted to supplement candidate classes for reducing the probability of missing the true class, according to the linguistic context or the classification confusion matrix. These techniques, however, have not been evaluated in integrated segmentation-and-recognition.

3. SYSTEM OVERVIEW

This study focuses on the recognition of text lines, which are assumed to have been segmented externally. The text lines in our database have been segmented and annotated at character level [19].

Fig. 2 shows the block diagram of our system for text line recognition. First, the input text line image is over-segmented into a sequence of primitive segments (Fig. 3a) using the connected component-based method [5]. Consecutive primitive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice (Fig. 3b). After that, each candidate pattern is classified to assign a number of candidate character classes, and all the candidate patterns in a candidate segmentation path generates a character candidate lattice (Fig. 3c). If a word level language model is used, each sequence of candidate characters is matched with a word lexicon to segment into candidate words, forming a word candidate lattice (Fig. 3d). All of these character (or word) candidate lattices are merged to construct the segmentation-recognition lattice of text line image. Each path in this lattice is constructed by a character sequence paired with a candidate pattern sequence, and this path is called a candidate segmentation recognition path. Finally, the task of string recognition is to find the optimal path in this segmentation-recognition lattice. Considering that the text lines are segmented from text pages, we utilize the linguistic dependency between consecutive lines to improve the recognition accuracy by concatenating multiple top-rank recognition results of the previous line to the current line for recognition.



(a) (b) (c) (d)

Fig.3.(a) Over-segmentation to a sequence of primitive segments (each is bounded by a small box), (b) segmentation candidate lattice of the arrowed part of (a), (c) character candidate lattice of the thick path in (b), (d) word candidate lattice of (c).

4. MODELS OF CONTEXTS

The path evaluation functions (11)-(14) entail the estimation of the context models and the combining weights.

4.1 Geometric Models:

Considering that Kannada texts mix with alphanumeric characters and punctuation marks and different characters show distinct outline features (e.g., size, position, aspect ratio, and within-character gap), we design two class dependent geometric models, namely, single-character geometry (unary geometric model) and between-character geometry (binary geometric model), respectively. In addition, two class-independent geometric models are designed to indicate whether a candidate pattern is a valid character or not, and whether a gap is a between-character gap or not, respectively. The four geometric models (unary and binary class-dependent, unary and binary class-independent) are abbreviated as “ucg,” “bcg,” “uig,” and “big,” respectively, and have been used successfully in transcript mapping of handwritten Kannada documents [5]. To build geometric models, we extract features for unary and binary geometry from the bounding boxes and profiles of a candidate character pattern, and of two consecutive character patterns, respectively [25].

5. PROPOSED SYSTEM

This experiment focuses on the recognition of text lines, which are assumed to have been segmented externally. For the convenience of academic research and benchmarking, the text lines in our database have been segmented and annotated at character level. First, the input text line image is over segmented into a sequence of primitive segments using the connected component-based method. Consecutive primitive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice. After that, each candidate pattern is classified to assign a number of candidate character classes, and all the candidate patterns in a candidate segmentation path generate a character candidate lattice.

5.1 Path search Algorithm

On defining a score for each path in the segmentation recognition lattice, the next issue is how to efficiently find the path of maximum score. In addition, to alleviate the loss that the candidate classes assigned by character classifier do not contain the true class, we propose an augmentation technique to supplement candidate classes in the lattice.

Search Algorithm

If the segmentation-recognition path is evaluated by the accumulated score of character pattern width, it satisfies the principle of optimality, and the optimal path with maximum score can be found by dynamic programming. A simple strategy of beam search is to retain the multiple top-rank partial paths ending at each primitive segment. This simple strategy, though it works efficiently, is too rough, particularly when high-order context models are used.

After over-segmentation, the text line image is represented as a sequence of primitive segments. A candidate pattern composed of k consecutive segments and ending at the ith segment is denoted by (i, k). A node in the search space is represented as a quadruple SN = {CP, CC, AS, PN}, where SN denotes a search node, CP is a candidate pattern, CC is a candidate character of CP, and AS is the accumulated score from the root node (calculated by (11)-(14), where m is the length of the current partial path), and PN is a pointer to the parent node of SN. All nodes are stored in a list named LIST to backtrack the final path. The refined beam search process is described in detail as follows,

Refined Beam Search in frame-synchronous fashion:

1. Initialize the first search node (i.e., the root) of LIST, SN_0 = {null, null, 0, null}, set i = 1.
2. Generate nodes of CP = (i , k) over k, i – k ≥0, k ≤ K, K is the maximum number of segments to be concatenated). For each CP, the CN (Candidate Number) candidate characters are assigned by the character classifier. In total, at most K × CN nodes are generated.
3. Link to parent nodes for current nodes (CP = (i , k), CC=c_{i-k}).For multiple such parent nodes (CP'=(i-k,k'),CC'=$c_{i-k,k'}$), the current node generates multiple copies, each linked to a respective parent node (PN) and associated
4. to an accumulated score (AS). In this, only the node with maximum AS over (k',$c_{i-k,k'}$) is retained.
5. Sort the retained nodes in above in decreasing order according to AS over (k, $c_{i,k}$), and the leading BW (Beam Width) nodes are retained and added to **LIST**, while the others are pruned to accelerate search.

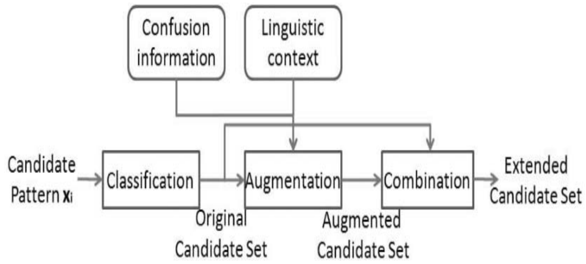


Fig.5. Diagram of candidate character augmentation

6. Set i=i+1, back to Step 2 and iterate until the last primitive segment is reached (such nodes called terminal nodes).
7. Backtrack the terminal node in **LIST** of maximum score along the element PN, and obtain the result character string.

We can see that if BW = K × CN, the above algorithm guarantees finding the optimal path for context models up to order 2 when the principle of optimality is satisfied, i.e., it is equivalent to DP. For context models of order 3 (e.g., character trigram) or higher, it does not guarantee finding the optimal path but significantly accelerates search compared to DP. Further, if BW = K × CN, the search procedure is further accelerated. Compared to simple beam search, the two-step pruning strategy in the refined beam search algorithm [25] has at least two advantages: 1) The first step pruning (in Step 3) observes the principle of optimality; 2) sorting the nodes has lower complexity. If we use word level n-grams, the search process works on a word candidate lattice, which is constructed from character lattice by combining several consecutive characters according to the word lexicon. So, search in the word candidate lattice is very complex [25]. To accelerate this search process, we first prune the original character lattice using the above character search process (many nodes are pruned in Steps 3 and 4), then use it to construct a succinct word lattice.

6. RESULTS

Experimental results are as follows:

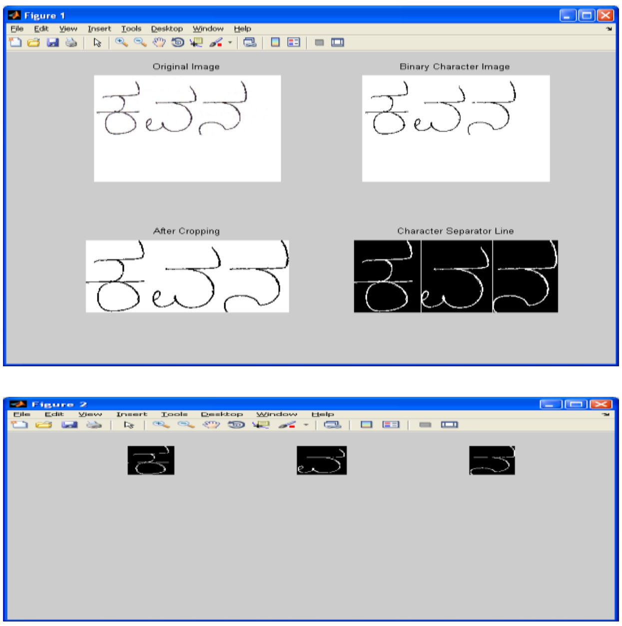


Fig. 4: Kannada Word Processing

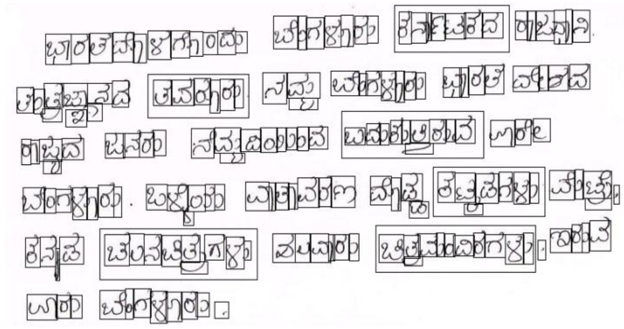


Fig. 5: Kannada Text Processing

7. CONCLUSION

This proposed work presented an approach for handwritten Kannada text recognition under the character over-segmentation and candidate path search, character augmentation [26] framework. It Recognizes the handwritten Kannada text whenever there is multiple contexts in the text pattern and it is independent of size, slant, orientation, word gap and translation. We introduce the paths from the Bayesian decision view by combining multiple contexts, including the character classification scores, geometric contexts. The combining weights of path evaluation function are optimized by a string recognition objective, namely, the Maximum Character Accuracy criterion. In path search, we use a refined beam search algorithm to improve the accuracy, efficiency and better readability.

The analysis of recognition errors indicates that the future work is needed to improve the character over segmentation, character classification, and path evaluation. The objective of over segmentation is to improve the tradeoff between the number of splitting characters and the accuracy of separating characters at their boundaries. The objective of character classification is to improve the classification accuracy and the tradeoff between the number of candidate classes and the readability of character true class. For path evaluation, both the geometric model and the language model deserve elaboration. Particularly, our experimental results show that mismatch of language model and text domain leads to inferior recognition performance. Hence, this system is used to recognize the handwritten Kannada postal codes, bank checks, and application forms.

8. REFERENCES

[1] Statistical Texture Features based Handwritten and Printed Text Classification in South Indian Documents Mallikarjun Hangargea, K.C. Santosh, Srikanth Doddamania, Rajmohan Pardeshi

[2] Divide and Conquer Technique in Online Handwritten Kannada Character Recognition M. Mahadeva Prasad Dept. of Electronics Mysore University P.G. Centre Hassan, INDIA.prasada9@gmail.com M. Sukumar Dept. of Instrumentation Technology S. J. College of Engineering Mysore, INDIA.m_sukumar@rediffmail.com A.G. Ramakrishnan Dept. of Electrical Engineering Indian Institute of Science Bangalore, INDIA. ramkiag@ee.iisc.ernet.in

[3] Recognition of isolated handwritten Kannada vowels Sangame S.K.1, Ramteke R.J.1, Rajkumar Benne2.

[4] K. Kuhnke, L. Simoncini, Z. M. Kovacs-V, A system for machine-written and hand-written character distinction,

in: Proceedings of International Conference on Document Analysis and Recognition, 1995, pp. 811 – 814.

[5] T. N. Tan, “Written language recognition based on Texture Analysis”, Proc. of ICIP 96, Lausanne, Switz, vol. 2, pp. 185-188, 1996.

[6] B.V. Dhandra, Mallikarjun Hangarge, Gururaj Mukarambi, ”Spatial Features for Handwritten Kannada and English Character Recognition System”, Special Issue on RTIPPR-10, International Journal of Computer Applications (IJCA).

[7] R.-W. Dai, C.-L. Liu and B.-H. Xiao, “Kannada Character Recognition: History, Status and Prospects,” Frontiers of Computer Science in China, vol. 1, no. 2, pp. 126-136, 2007

[8] M. Wuthrich, M. Liwicki, A. Fischer, E. Indermuhle, H. Bunke, G. Viehhauser, and M. Stolz, “Language Model Integration for the Recognition of Handwritten Medieval Documents,” Proc. 10th Int’l Conf. Document Analysis and Recognition, pp. 211-215, July 2009.

[9] M.-Y. Chen, A. Kundu, and S.N. Srihari, “Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition,” IEEE Trans. Image Processing, vol. 4, no. 12, pp. 1675-1688, Dec. 1995.

[10] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, “Online and Offline Handwritten Kannada Character Recognition: Benchmarking on New Databases,” Proc. Second CJK Joint Workshop Pattern Recognition, Oct. 2010.

[11] H.-S. Tang, E. Augustin, C.Y. Suen, O. Baret, and M. Cheriet, “Spiral Recognition Methodology and Its Application for Recognition of Kannada Bank Checks,” Proc. Ninth Int’l Workshop Frontiers in Handwriting Recognition, pp. 263-268, Oct. 2004

[12] S. Quiniou, M. Cheriet, and E. Anquetil, “Handling Out-of- Vocabulary Words and Recognition Errors Based on Word Linguistic Context for Handwritten Sentence Recognition,” Proc. 10th Int’l Conf. Document Analysis and Recognition, pp. 466-470, July 2009.

[13] M. Cheriet, N. Kharma, C.-L. Liu, and C.Y. Suen, Character Recognition Systems: A Guide for Students and Practitioners. John Wiley & Sons, Inc., 2007. [17] H. Murase, “Online Recognition of Free-Format Japanese Handwritings,” Proc. Ninth Int’l Conf. Pattern Recognition, pp. 1143-1147.

[14] X.-D. Zhou, C.-L. Liu, and M. Nakagawa, “Online Handwritten Japanese Character String Recognition Using Conditional Random Fields,” Proc. 10th Int’l Conf. Document Analysis and Recognition, pp. 521-525, July 2009.

[15] L.Y. Tseng and R.C. Chen, “Segmenting Handwritten Kannada Characters Based on Heuristic Merging of Stroke Bounding Boxes and Dynamic Programming,” Pattern Recognition Letters, vol. 19, no. 10, pp. 963-973, Aug. 1998.

[16] Z. Liang and P. Shi, “A Metasynthetic Approach for Segmenting Handwritten Kannada Character Strings,” Pattern Recognition Letters, vol. 26, no. 10, pp. 1498-1511, July 2005.

[17] C.-L. Liu, “Handwritten Kannada Character Recognition: Effects of Shape Normalization and Feature Extraction,” Proc. Conf. Arabic and Kannada Handwriting Recognition, S. Jaeger and D. Doermann, eds., pp. 104-128, 2008.

[18] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, “Modified Quadratic Discriminant Functions and the Application to Kannada Character Recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 9, no. 1, pp. 149-153, Jan. 1987.

[19] C.-L. Liu and M. Nakagawa, “Evaluation of Prototype Learning Algorithms for Nearest Neighbor Classifier in Application to Handwritten Character Recognition,” Pattern Recognition, vol. 34, no. 3, pp. 601-615, Mar. 2001.1480 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, http://ieeexploreprojects.blogspot.com

[20] Y. Jiang, X. Ding, Q. Fu, and Z. Ren, “Context Driven Kannada String Segmentation and Recognition,” Proc. Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR Int’l Workshops, pp. 127- 135, 2006.

[21] Y.X. Li, C.L. Tan, and X.Q. Ding, “A Hybrid Post-Processing System for Offline Handwritten Kannada Script Recognition,”Pattern Analysis and Applications, vol. 8, no. 3, pp. 272-286, 2005

[22] Q.-F. Wang, F. Yin, and C.-L. Liu, “Integrating Language Model in Handwritten Kannada Text Recognition,” Proc. 10th Int’l Conf. Document Analysis and Recognition, pp. 1036-1040, July 2009.

[23] M. Nakagawa, B. Zhu, and M. Onuma, “A Model of On-Line Handwritten Japanese Text Recognition Free from Line Direction and Writing Format Constraints,” IEICE Trans. Information and Systems, vol. 88, no. 8, pp. 1815-1822, Aug. 2005..

[24] L.Y. Tseng and R.C. Chen, “Segmenting Handwritten Kannada Characters Based on Heuristic Merging of Stroke Bounding Boxes and Dynamic Programming,” Pattern Recognition Letters, vol. 19, no. 10, pp. 963-973, Aug. 1998.

[25] Qiu-Feng Wang, Fei Yin, and Cheng-Lin Liu, “Handwritten Kannada Text Recognition by Integrating Multiple Contexts”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 8, AUGUST 2012.