

Unrestricted Kannada Online Handwritten Akshara Recognition using SDTW

Rituraj Kunwar, Mohan P., Shashikiran K, A. G. Ramakrishnan
MILE Lab, Department of Electrical Engineering, IISc, Bangalore, India.
{kunwar.rituraj, mohan.iisc, shashi.reach, agrkrish} @gmail.com

Abstract—In this paper, we present an unrestricted Kannada online handwritten character recognizer which is viable for real time applications. It handles Kannada and Indo-Arabic numerals, punctuation marks and special symbols like \$, &, # etc, apart from all the aksharas of the Kannada script. The dataset used has handwriting of 69 people from four different locations, making the recognition writer independent. It was found that for the DTW classifier, using smoothed first derivatives as features, enhanced the performance to 89% as compared to preprocessed co-ordinates which gave 85%, but was too inefficient in terms of time. To overcome this, we used Statistical Dynamic Time Warping (SDTW) and achieved 46 times faster classification with comparable accuracy i.e. 88%, making it fast enough for practical applications. The accuracies reported are raw symbol recognition results from the classifier. Thus, there is good scope of improvement in actual applications. Where domain constraints such as fixed vocabulary, language models and post processing can be employed. A working demo is also available on tablet PC for recognition of Kannada words.

I. INTRODUCTION

Rapid development in technology has made handheld devices very popular and in coming days with increase in demand, it will be affordable too. This will lead to handwriting recognition being an alternative to the keyboard as an input device, along with, may be, speech recognition. Data entry using pen is a natural form of interface. Various techniques have been explored for handwriting recognition and efficient practical applications exist for English. However, there is no such system for Indian languages so far. Majority of the research reported for Indian languages have either dealt with a subset of characters such as only the base characters or the numerals, or approaches based on limited vocabulary lexicon based recognizers using HMM. While the above approaches have their own applicability, when it comes to unlimited vocabulary recognition involving proper names, addresses, etc. the above approaches simply cannot be used at all. In this paper, we aim to develop a recognition system that handles the complete set of Kannada aksharas, including the Kannada numerals and Indo-Arabic numerals, besides other symbols.

DTW is a technique that is used for elastic matching of two signals by warping one axis of one or both signals. This method has been efficiently used in speech processing, sign language analysis, online and offline character recognition etc. In DTW, one axis is warped to explain variability in the other axis.

In this method, multiple templates need to be maintained corresponding to a character for computing DTW distance to cover all the variations in writing that character. Moreover, DTW distance calculation is computationally expensive. While testing, it is required to calculate DTW distances to all of these templates and assign the class label of the template nearest to test pattern. Therefore, to avoid the above computationally expensive operations, one model per class was trained which represents all the transitions made while calculating DTW distance from each of the training data to the model. Claus Bahlmann et al proposed one such approach, namely, Statistical Dynamic Time Warping (SDTW). With this approach, we can overcome the limitations of NN with DTW at the same time retaining the quality to consider local temporal information in data.

II. SURVEY OF KANNADA OHR

A. Kannada Script:

Kannada is the official language of the South Indian state of Karnataka. It has its own script derived from Bramhi script. Kannada script has a base set of 52 characters, comprising 16 vowels and 36 consonants. Further there are distinct symbols that modify the base consonants, called consonant and vowel modifiers. The number of these modifiers is the same as that of the base characters. The characters called aksharas are formed by graphically combining the symbols corresponding to consonants, consonant modifiers (optional) and vowel modifiers using well defined rules of combination. Therefore, the number of theoretically possible combinations of Kannada characters are as follows:

Number of vowels is 16

Number of possible consonant-vowel combinations:
 $36 \times 16 = 576$

Number of possible consonant-consonant-vowel combinations:
 $36 \times 36 \times 16 = 20736$

While designing a character recognition system, if we consider each akshara as a separate class, the number of classes becomes prohibitively high.

However in Kannada, all consonant modifiers are written separately from the base character and at least some part of all the strokes of consonant modifiers will lie below the

base character as in Fig.1. So we considered the consonant modifiers as separate classes. This reduces $36 \times 36 \times 16$ C-C-V combinations (or possible classes) to 36×16 C-V combinations of the base character and additional 36 classes of the consonant modifiers.

Similarly, some of the vowel modifiers are also written

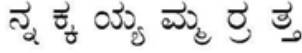


Fig. 1. Consonant modifiers lie below the base character

separately from the base character as shown in Fig. 2 . Thus, considering these as separate classes, we reduced the total number of classes further.

In all, we reduced the total number of classes to 295

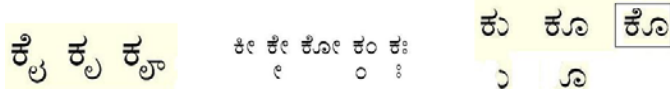


Fig. 2. Examples of some of the vowel modifiers written separately from the base character, which could therefore be segmented from the character complex (akshara) and recognised separately.

including Indo-Arabic numerals, Kannada numerals and punctuation marks. By recognizing these symbols, we cover the whole of the Kannada character set. One of the applications we have in mind for our recognizer is that of a form-filling application, which necessarily has names, numbers, punctuation marks and special symbols. Hence, we include all of the above as pattern classes.

B. Status of Kannada OHR:

Work on Kannada OCR and OHR have been few and far between. One of the few works reported in Kannada character OCR was by P.S. Sastry et. al. [9] which was font and size independent with reasonably good performance between 80 & 86%. Another work done on OCR of Kannada character was by Vijay Kumar et. al. [8] using neural network with accuracy approximately 95%. Similar work on Kannada offline numeral handwritten recognition was conducted by S.V. Rajashekaradahya et. al. [11] with recognition accuracy of around 95%. One of the works on Kannada character OHR is by S.R. Kunte et. al.[7] using wavelet features and neural network as classifier, reporting an accuracy of 95%. But in none of the above cases, details of the dataset used are given, rendering the different works incomparable.

To the best of our knowledge this is the first work in Kannada OHR that covers the complete character set of Kannada: base characters, vowel modifiers, ottus and numerals (both Kannada and Indo-arabic). It is difficult to infer few details from the previous works in Kannada OHR as to whether they are writer dependent or independent and

how efficient they are in terms of time.

In the present work, we have addressed these issues. We have trained our Kannada OHR engine with data from 40 users and made it writer independent. In spite of covering all the characters of Kannada, symbols, etc. We managed to have the number of classes manageable by exploiting the nature of the script in sub-section A, above. It is evident from our results shown in Table 1 that our above approach can be applied in real time applications.

III. CLASSIFICATION EXPERIMENTS

Fig.3 shows the basic building blocks of our OHR.

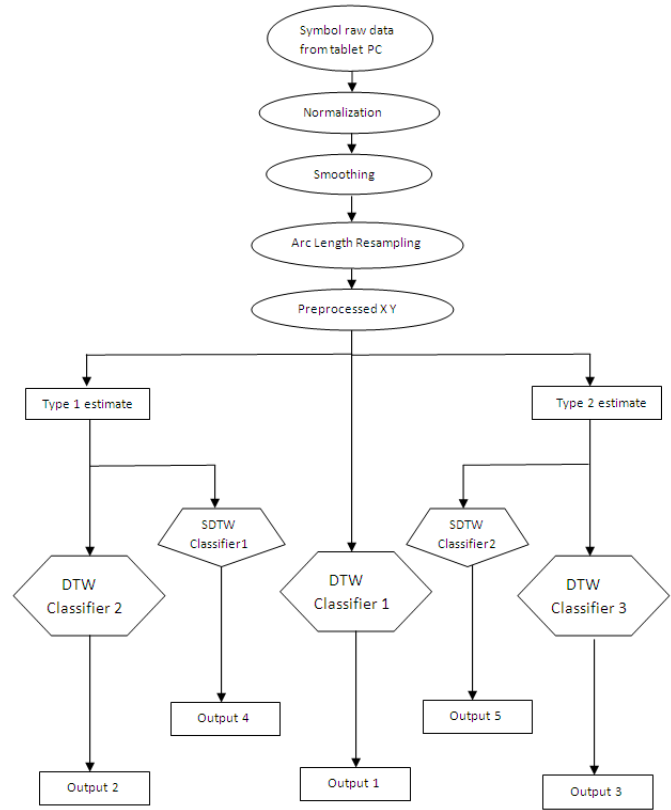


Fig. 3. Block schematic of Kannada OHR experiments

Raw data captured from the Tablet PC needs to be smoothed in order to remove any noise, which may be present due to the erratic movement of hand while writing. In addition, preprocessing involves, normalization and resampling based on equi-arc length.

A. Dynamic Time Warping (DTW):

This is a technique useful for intuitive matching of two patterns having equal or unequal length [1]. Fig. 4 shows two patterns (test 'T' and reference 'R') of different lengths (different samples of the character /u/ in Kannada) matched by

DTW technique. In the matching process, a cumulative cost matrix is created as shown in Fig. 4 that shows which point(s) of the reference pattern 'R' match best with which point(s) of test pattern 'T'.

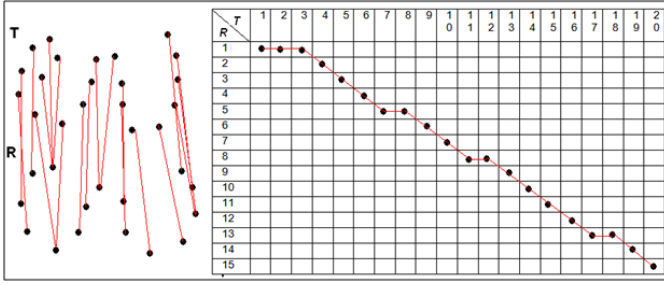


Fig. 4. Transitions between sample points of reference and test patterns in DTW matrix

Consider two sequences $R = (r^1, r^2, r^3, r^4, \dots, r^J)$ and $T = (t^1, t^2, t^3, t^4, \dots, t^K)$ where $(r^j, t^k) \in R^d$. Let the warping path be $\phi = (\phi(1), \phi(2), \phi(3), \dots, \phi(N))$ with $\phi(n) = (r^j, t^k)$ which gives the details of alignment of pattern R to T.

1) Constraints on warping path:

a) The first and last points of pattern T are matched with the first and last points of pattern R, respectively. i.e., $\phi(1) = (1, 1)$ and $\phi(N) = (J, K)$, where N is the total instances in the warping path.

b) $\phi(m) = (\alpha, \beta)$, $\phi(m-1) = (\alpha', \beta')$ where $0 \leq (\alpha - \alpha') \leq 1$ and $0 \leq (\beta - \beta') \leq 1$.

The above condition makes sure that the path will be moving either right by one step or down by one or diagonally below towards right by one step.

2)Steps followed:

1) A distance matrix of dimension $J * K$ is created, which contains Euclidean distance of every point of R with every point of T whose elements are $d(i,j) = \text{EuclideanDist}(r^i, t^j)$

2) A cumulative cost matrix of dimension $J * K$ is generated, whose elements are calculated as follows $c(i,j) = d(i,j) + \min(c(i-1,j), c(i,j-1), c(i-1,j-1))$;

3) Warping cost = $\text{DTW}(R, T) = c(J, K)$.

4) Warping path could be found out using dynamic programming technique as is done in "Viterbi Algorithm" to find Viterbi path.

3) Limitations of x, y as features in DTW:

The Euclidean distance metric that this technique uses is sensitive to the y value of the point; it does not differentiate between the points corresponding to the rising and falling slopes, if the difference in y values is the same. For this reason, one large subsection of one pattern may get matched

to one particular point in the other pattern. This leads to unintuitive matching, which is undesirable. For details, refer [1].

4) First Derivatives as feature:

First Derivative (Estimate 1): In this method, the derivative at the current point is estimated using the formula given below:

$$X'(j) = \frac{\sum_1^2 i \times (x(j+i) - x(j-i))}{2 \times \sum_1^2 i^2}, \quad Y'(j) = \frac{\sum_1^2 i \times (y(j+i) - y(j-i))}{2 \times \sum_1^2 i^2}$$

Since the above formula cannot be estimated for the 1st, 2nd, last and 2nd last points, their values are calculated as shown below:

$$X'(1) = X'(2) = X'(3); \quad Y'(1) = Y'(2) = Y'(3) \\ X'(l) = X'(l-1) = X'(l-2); \quad Y'(l) = Y'(l-1) = Y'(l-2)$$

where $1 \leq i \leq L$ where L = number of points in the pattern. Warping cost is evaluated as explained above with features $X'(j)$ $Y'(j)$.

First Derivative (Estimate 2): Another estimate of derivatives of x and y at each point was suggested as a feature in [1]. The estimated derivative of x and y at the point j in a pattern could be calculated using the formulae given below:

$$X'(j) = \frac{x(j) - x(j-1) + \frac{x(j+1) - x(j-1)}{2}}{2}, \\ Y'(j) = \frac{y(j) - y(j-1) + \frac{y(j+1) - y(j-1)}{2}}{2}$$

Since the above formula cannot be estimated for the first and last points, their values are assumed to be the same as those of second and penultimate points, respectively:

$$X'(1) = X'(2); \quad Y'(1) = Y'(2); \quad X'(l) = X'(l-1); \quad Y'(l) = Y'(l-1)$$

where $1 \leq i \leq L$ where L = number of points in the pattern. Warping cost is evaluated as explained above with features $X'(j)$, $Y'(j)$.

Pseudo code for Testing:

```

Loop1 ClassReference = 1:TotalNoOfClass
  Loop2 SampleReference=1:TotalNoOfReferenceSample
    DTWDistanceMat[ClassReference][SampleReference]=
      DTWDistance(ClassTest,SampleTest,ClassReference,
        SampleReference);
  End
End
Loop3 ClassReference=1:TotalNoOfClass
  Distance[ClassReference]=
    sumRowElements(DTWDistanceMat);
End
ClassToTestSampleBelong = IndexMinimum (Distance);

```

B. Statistical Dynamic Time Warping (SDTW):

Limitation of DTW: In DTW, we need to maintain a minimum number of templates for each class to cover all the possible variations in writing the characters. So, while testing, it is required to calculate the DTW distance to all of these templates and assign the class label of the template nearest to the test pattern. This process is computationally expensive. To overcome this, Claus Bahlmann et. al. [4] have suggested Statistical DTW (SDTW).

SDTW: In SDTW, a reference character is not represented by a sequence of feature vectors as given in [4] and [12]. Instead it is represented by a sequence $Q = (Q^1, Q^2, Q^3, Q^{lq})$ of statistical quantities (states), as shown in Fig 5. These statistical quantities include

1) Discrete probabilities say $\alpha^j : \Omega \rightarrow [0, 1]$ for statistical modeling of transitions $\Delta\phi \in \Omega$ reaching the sequence's state j . i.e. state transition probabilities are defined by

$$\alpha_j(\Delta\phi) = P(\Delta\phi = \phi(n) - \phi(n-1) | \phi_R(n) = j) \quad , \quad \Delta\phi \in \Omega.$$

In the special case of $n = j = 1$,

$$\alpha_1((1,1)) = P(\Delta\phi(1) = \phi(1) - \phi(0) | \phi_R(1) = 1) = 1.$$

2) A continuous probability density function $\beta_j : R^d \rightarrow R$ that models the feature distribution at sequence's state j . In our work we modeled β_j by a unimodal, multivariate Gaussian distribution i.e.

$$\beta_j(x) = P(x | \phi_R(n) = j) = N_{\mu_j, \Sigma_j}(x) = \frac{1}{(2\pi)^d |\Sigma_j|} \exp\left(-\frac{1}{2}(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)\right)$$

In some situations, the probability $\alpha'_j(\Delta\phi)$ that a transition $\Delta\phi$ emerges from the sequence's state j is also required and is related to $\alpha_j(\Delta\phi)$ as

$$\alpha_j(\Delta\phi) = \alpha'_{j-\Delta\phi_R}(\Delta\phi) \quad \text{and} \quad \sum_{\Delta\phi} \alpha'_j(\Delta\phi)$$

While testing, the SDTW distance of test pattern to the reference model of each class is computed and the test pattern is assigned the label of the class giving minimum SDTW distance. Here, the definition of SDTW distance is somewhat different from that of DTW and is given by

$$SD_\phi * (T, Q) = \min_\phi SD_\phi(T, Q)$$

where $SD_\phi(T, Q)$ is statistical warping distance between pattern T and model Q , with Φ as alignment path and is given by

$$SD_\phi(T, Q) = \sum_{i=1}^N (d(t_{\phi_t(i)}, Q_{\phi_q(i)}) - \log_e \alpha_{\phi_q(i)}(\Delta\phi(i)))$$

where

$$d(t_i, Q_j) = 0.5((t_i - \mu_j)^T \Sigma_j^{-1} (t_i - \mu_j) + d \log_e(2\pi) + \log_e |\Sigma_j|)$$

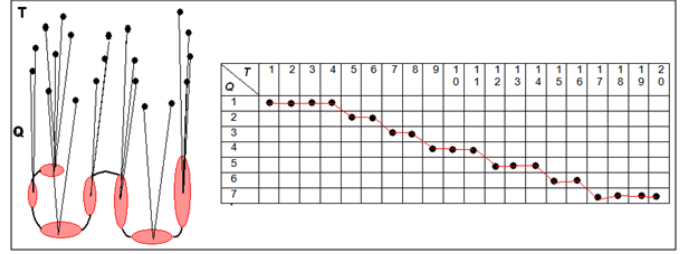


Fig. 5. Transitions between states in SDTW

Fig.5 shows how the matching takes place between the reference model and test pattern. The matrix in Fig.5 shows the SDTW path (path of best SDTW matching).

It can be observed that SDTW distance is the negative log state optimized likelihood of pattern T generated by the model Q , with optimal state sequence Φ^* given by the Viterbi algorithm. So, models in SDTW frame work are similar to HMMs [2] [3] of particular type with state prior probabilities $\pi = (1, 0, 0, \dots, 0)^T$ and are of left to right models with step size of at most 1 and with null transitions (transitions that allow change in state without observation change i.e. transitions $(0,1)$ in Ω). So the models in SDTW frame work can be trained by algorithms used for training HMMs. In our work, we used segmental K-means algorithm [3] for training SDTW model parameters.

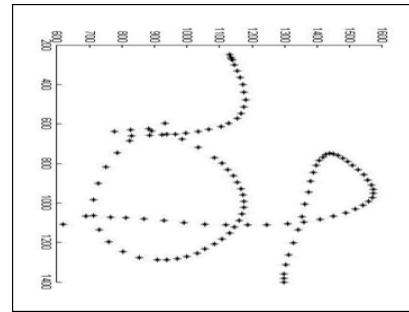


Fig. 6. An example raw character data from Tablet PC

```
.VERSION 1.0
.HIERARCHY CHARACTER
.COORD X Y T
.X_POINTS_PER_INCH 2500
.Y_POINTS_PER_INCH 2500
.POINTS_PER_SECOND 120
.SEGMENT CHARACTER 0 OK "000t01.txt"
.PEN_DOWN
935 523 0
935 523 0
935 523 0
935 520 0
.PEN_UP
.PEN_DOWN
1004 455 0
1019 455 0
1032 461 0
1044 474 0
1053 486 0
.PEN_UP
```

Fig. 7. Format of raw data from Tablet PC

IV. DATA COLLECTION FOR OUR EXPERIMENTS

MILE lab Kannada database for conducting the experiment has been created by collecting data from 69 different writers so that recognition engine could be trained with different styles of handwriting in turn making the OHR engine writer independent. Each writer wrote all the 295 symbols of the dataset.

Writers for the data collection were meticulously chosen; only those who regularly write Kannada were included. Justification for this constraint is that we need to capture the regular style of Kannada handwriting, which includes local temporal information.

OHR system requires a transducer based system to capture handwriting. Transducer captures handwriting by capturing the pen's movement on the screen in terms of x and y co-ordinates. Sequences of x , y co-ordinates are sampled at equal intervals of time along with PEN_DOWN and PEN_UP information. User writes with an electronic pen on the electrostatic pressure sensitive writing surface of a tablet PC. The pattern entered by user is displayed on the surface by means of digital ink and it provides interaction between the user and system. A sample raw data from tablet PC is shown in Fig.6 and its format is shown in Fig.7.

V. RESULTS

As seen from *Table 1.*, using the two types of derivatives in a DTW classifier, though reduces the recognition speed by a sec(due to the additional computation of the derivatives), it enhances the accuracy by 1% and 4% for estimate 1 and estimate 2 derivatives, respectively. Using SDTW, increases the speed by 46 times with a marginal reduction in accuracy.

All experiments were conducted on the Kannada Database of Medical Intelligence and Language Engineering (MILE) Lab, IISc. The results are shown in Table 1. Kannada dataset has 295 classes where each class represents a group of strokes. Each class has been trained and tested with 40 and 29 samples files, respectively.

Table 1. Writer independent recognition Performance Comparison of DTW and SDTW with different estimates of first derivative (No. of classes: 295; No. of training and testing samples/class: 40 and 29, respectively).

Classifier	Features used	Symbol Accuracy (%)	Recognition time/sample (sec)
Dynamic Time Warping (DTW)	Preprocessed X,Y co-ordinates	85.2	22
DTW	1 st derivatives of X,Y co-ordinates (Estimate 1)	86.0	23
DTW	1 st derivatives of X,Y co-ordinates (Estimate 2)	89.0	23
Statistical DTW	Preprocessed X,Y co-ordinates and 1 st derivatives of X,Y co-ordinates (Estimate 1)	85.2	0.5
Statistical DTW	Preprocessed X,Y co-ordinates and 1 st derivatives of X,Y co-ordinates (Estimate 2)	87.9	0.5

VI. CONCLUSION

We have demonstrated the effectiveness of first derivative Estimate 2 feature [1] in classifying Kannada handwritten characters. To our best knowledge, this is the first work reported in Kannada which deals with all combinations of consonants and vowels, punctuations, Kannada and Indo-Arabic numerals. The accuracies reported are raw symbol recognition results from the classifier and do not involve the use of any language models or postprocessing.

VII. ACKNOWLEDGMENT

We thank Mr. Venkatesh, Ms. Nethra Nayak, Ms. Archana C.P and Mr. Suresh Sundram for their efforts which made these experiments possible. Special thanks to Technology Development for Indian Languages (TDIL), Dept of Information Technology (DIT) for funding this research. We thank Vijaya college, Lakshmi Ammani College and Seshadripuram High School, all in Bangalore, for contributing to the dataset.

REFERENCES

- [1] E. Keogh and M. Pazzani, "Derivative dynamic time warping," First SIAM International Conference on Data Mining (SDM'2001), Chicago, USA, 2001.
- [2] Lawrence R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition", Proc. of the IEEE, Vol.77, No.2, pp 257-286, February 1989.
- [3] Rakesh Dugad and U. B. Desai "A tutorial on hidden markov models" SPANN- 96.1 May 1996.
- [4] Clauss Bahlmann and Hans Burkhardt , "The writer independent online handwriting recognition system Frog On Hand and cluster generative statistical dynamic time warping", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, No. 3, pp 299-310, March 2004.
- [5] Scott D. Connell, Anil K. Jain, "Template based online character recognition," Pattern Recognition, 34(2001), pp 1-14.
- [6] Kruskal, J. B., and Liberman, M. (1983) The symmetric time-warping problem: from continuous to discrete. Reprinted in Sankoff, D. and Kruskal, J., (eds.) (1999) Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. CSLI Publications, Stanford, pp. 125-161.
- [7] S. R. Kunte and Sudhakar Samuel. Wavelet features based online recognition of handwritten kannada characters. JI. Visualization Society of Japan, 20. (2000), 417-420.
- [8] B. Vijay Kumar and A.G. Ramakrishnan, Machine recognition of printed kannada text, Proc. 5th International workshop on document analysis and systems, 2002, 37-48.
- [9] 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, No. 1, 2002, 35-58.
- [10] Niranjana Joshi , G. Sita , A. G. Ramakrishnan , Sriganesh Madhvanath, Comparison of elastic matching algorithms for online tamil handwritten character recognition, Proc ninth international workshop on frontiers in handwriting recognition, p.444-449, October 26-29, 2004
- [11] S.V Rajashekararadhya and P. Vanaja Ranjan "Handwritten numeral recognition of three popular south Indian scripts: A novel approach:", Proc second international conference on information processing ICIP, pp.162-167, 2008.
- [12] Mohan P. "SVM and SDTW based online handwritten recognition of Kannada words" M.E Project Report, Dept. of EE, IISc, 2008.