Accuracy Augmentation of Tamil OCR Using Algorithm Fusion

R.Jagadeesh Kannan R. Prabhakar

RMK Engg College, Chennai, India. CIT, Coimbatore, India

Summary
The need for OCR arises in the context of digitizing Tamil documents from the ancient and old era to the latest, which helps in sharing the data through the Internet. Tamil, which is a south Indian language, is one of the oldest languages in the world. Even there are few works going on in Tamil OCR, the accuracy of the approaches still remains a challenging area of research. Many of the works related to Tamil OCR have not concentrated or dealt enough with the accuracy parameter. Our work is contributed to increase the performance of Tamil OCR. We have chosen two algorithms to be fused to get the advantage of both the algorithms. Fusing the algorithms yields efficiency of OCR conversion. Improvement in accuracy is proven through experimental results discussed in this paper.

Key words:
Optical Character Recognition (OCR), Hidden Markov Models (HMM), Support Vector Machines (SVM), Radial Basis Function Neural Network (RBF-NN).

1. Introduction

With the increasing widespread use of computers in the business and other areas, more and more organizations are converting their paper documents into electronic documents that can be processed by computers. This also leads to the development of optical character recognition (OCR). Recognition of characters from document images is at the heart of any document image understanding system [1]. Optical Character Recognition (OCR) is a type of document image analysis where a scanned digital image that contains either machine printed or handwritten script is input into an OCR software engine and translating it into an editable machine readable digital text format (like ASCII text).

OCR is one of the most fascinating and challenging areas of pattern recognition with various practical applications. It can contribute immensely to the advancement of an automation process and can improve the interface between man and machine in many applications [2], [3]. Some applications of OCR are (1) Reading aid for the blind, (2) Automatic text entry into the computer for desktop publication, library cataloging, ledgering, etc. (3) Automatic reading for sorting of postal mail, bank cheques and other documents, (4) language processing etc. There are many uses for the output from an OCR engine and these are not limited to a full text representation online that exactly reproduces the original.

The most crucial stage in the process of Optical Character Recognition (OCR) [29] is that of recognizing the characters and classifying them. The other processes involved include preprocessing activities like binarization and skew estimations. It is followed by major phases like Segmentation and Feature Extraction. Among different branches of character recognition it is easier to recognize English alphabets and numerals than Tamil characters [6].

The need for OCR arises in the context of digitizing Tamil documents from the ancient and old era to the latest, which helps in sharing the data through the Internet. Tamil, which is a south Indian language, is one of the old languages in the world. It has been influenced by Sanskrit to a certain degree [4]. But Tamil is unrelated to the descendents of Sanskrit such as Hindi, Bengali and Gujarati. Most Tamil letters have circular shapes partially due to the fact that they were originally carved with needles on palm leaves, a technology that favored round shapes. Tamil script is used to write the Tamil language in Tamil Nadu, SriLanka, Singapore and parts of Malaysia, as well as to write minority languages such as Badaga. Tamil alphabet consists of 12 vowels, 18 consonants and one special character (AK). Vowels and consonants are combined to form composite letters, making a total of 247 different characters and some Sanskrit characters. The complete Tamil alphabet and composite character formations are given in [5]. The advantage of having a separate symbol for each vowel in composite character formations, there is a possibility to reduce the number of symbols used by the alphabet.

There are several methods available in the literature for the recognition of Tamil characters. They include the method proposed by Hewavitharana et al. [4], Chinnuswamy et al. [5], Suresh et al. [6], and Siromoney et al. [7], etc. We have proposed a new approach where we use Histogram Equalization and Gabor Filters to do image enhancement. Then Binarisation is applied on the enhanced image. Locally adaptive threshold method [22] is used for this process. Then Region of Interest [ROI] extraction is done using Morphological operations [22], [23]. Then character segmentation is done using Region probe algorithm.

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

Finally we have fused two pattern recognition algorithms which will result in better efficiency.

The rest of the paper is organized as follows. Section 2 explains our proposed Tamil OCR algorithm. Experimental results are given in section 3.Section 4 presents the conclusion.

2. Proposed Tamil OCR Algorithm

The accuracy or efficiency of OCR purely depends on the algorithm we deploy. The efficiency decreases when an algorithm fails to identify a character or if the algorithm detects an unrelated character. We have proposed a method where we can fuse two pattern recognition algorithms and evaluate the efficiency of OCR.

Before fusing, the scanned document is preprocessed. The steps in preprocessing involves

1) Histogram equalization and Gabor Filtering
2) Binarisation
3) ROI extraction
4) Region Probe Algorithm

2.1 Preprocessing Steps

This section describes the preprocessing steps of the scanned document in detail.

2.1.1 Histogram Equalization And Gabor Filetring

The scanned document is first applied to Histogram equalization and Gabor Filetring**.** Histogram equalization usually increases the local contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. Then the Gabor filter is applied to the scanned document obtained by the previous step by spatially convolving the image with the filter.

A Gaussian function [24] multiplied by a harmonic function defines the impulse response of the linear filter, the Gabor filter. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function.

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}+\psi)$$

Where,

$$x' = x\cos\theta + y\sin\theta$$

and

$$y' = -x\sin\theta + y\cos\theta$$

In this equation, λ represents the wavelength of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, Ψ is the phase offset, and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

2.1.2 Binarisation

The binarization process involves analyzing the grey-level value of each pixel in the enhanced image, and, if the value is greater than the global threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero.

2.1.3 Roi Extraction

We perform morphological opening on the grayscale or binary image with the structuring element. We also performed morphological closing on the grayscale or binary image resulting in closed image. The structuring element is a single structuring element object, as opposed to an array of objects for both open and close. Then as the result this approach throws away those leftmost, rightmost, uppermost and bottommost blocks out of the bound so as to get the tightly bounded region just containing the bound and inner area.

2.1.4 Region Probe Algorithm

After all the above process, the image is passed to the segmentation phase, where the image is decomposed into individual characters. For this we have used region probs algorithm.

2.2 Algorithm Fusion

Then we have fused two algorithms meaning that both the algorithms are taken into consideration. While fusing the two algorithms the following points are taken into consideration

1) If one algorithm fails to identify a character, another algorithm may support in identifying the character.
2) If one algorithm gives wrong character another may give a correct one.
3) The possibility for same wrong identification by both the algorithms is less.
4) If one algorithm gives wrong result the decision of choosing the correct result is done by neural network which is discussed later in the paper

We have chosen SVM, HMM for the fusion and discussed in the rest of the section.

2.3 Hidden Markov Model (Hmm)

Hidden Markov Models are suitable for handwriting recognition for a number of reasons [8]. The importance of HMMs in the area of speech recognition has been observed several ago [25]. In the meantime, HMMs have also been successfully applied to image pattern recognition problems such as shape classification [26] and face recognition [27]. HMMs qualify as suitable tool for cursive script recognition for a number a reasons. First, they are stochastic models that can cope with noise and pattern variations occurring is human handwriting. Next, the number of tokens representing an unknown input word may be of variable length.

Moreover, using an HMM-based approach, the segmentation problem, which is extremely difficult and error prone, can be avoided. This mean that the features extracted from an input word need not necessarily correspond with letters. Instead, they may be derived from part of one letter, or from several letters. Thus the operations carried out by an HMM are in some sense holistic, combining local feature interpretation with contextual knowledge. Finally, there are standard algorithms known from the literature for both training and recognition using HMMs. These algorithms are fast and can be implemented with reasonable effort. Kundu and Bahl built an HMM for the English language [28]. However, they require the input word being perfectly segmented into single characters.

The Hidden Markov Model is a straightforward generalization of ordinary probability distributions to the case of randomly generated sequences of discrete or continuous-valued events. A discrete density HMM produces strings $O = O_1 ... O_T$ of symbols form a finite alphabet $\{V_1,...,V_K\}$, while the continuous density version creates sequences of real-valued feature vectors $x \in IR^d$ .The generation of the observable outputs O_t of the model is controlled by a doubly stochastic process. At each time instant $t = 1,...,T$, the model is in one out of N possible states $\{S_1,...,S_N\}$. The state q_t taken by the model at time t is a random variable which depends only on the identity of its immediate predecessor state. According to this assumption the state distribution is completely determined by the parameters.

$$\pi_i = p(q_1 = s_i) \text{ and } a_{i,j} = p(q_t = s_j \mid q_{t-1} = s_i)$$

in other words, the vector $\pi = (\pi_1,...,\pi_N)^T$ of initial probabilities together with the $(N \times N) - matrix A = [a_{i,j}]$ of transition probabilities form a first-order Markov chain.

The actual state sequence taken by the model serves as a probabilistic trigger for the production of the output sequence. The q_t themselves, however, remain hidden to an observer of the random process. According to a second model assumption. The probability distribution of an output symbol O_t (or an output vector, respectively) depends solely on the identity of the present state q_t ; thus, the distribution parameters.

$b_{j,k} = b_j(v_k) = p(O_t = V_k \mid q_t = s_j)$ of a discrete density HMM constitute a $(N \times K)$ probability matrix $B = [b_{j,k}]$. Consequently, the behavior of an HMM with discrete output is entirely specified by the cardinality N of the state space, the alphabet size K , and a parameter array $\lambda = (\pi, A, B)$ of non-negative probabilities, obeying the $(1 + 2N)$ normalization condition

$$\sum_i \pi_i = \sum_j a_{i,j} = \sum_k b_{j,k} = 1.$$

Any of the state-dependent probability density functions (PDF) $b_j(x) = p(O_t = x \mid q_t = s_j)$ of an HMM with continuous output can be reasonably well approximated by a convex combination

$$b_j(x) = \sum_{k=1}^{K} b_{j,k} . g_{j,k}(x) = \sum_{k=1}^{K} b_{j,k} . N(x \mid \mu_{j,k}, \sum_{j,k})$$

of multivariate Gaussian PDFs. The huge amount of statistical parameters found in a continuous mixture HMM as defined above − the model includes estimates of a distribution mean $\mu_{j,k} \in IR^d$ and a covariance matrix $\sum_{j,k} \in IR^{d \times d}$ for each of $(N.K)$ index pairs can be drastically reduced if all state-dependent sets $\{g_{j,k} \mid K = 1,...,K\}$ of mixture components are pooled into one global collection of PDFs. The resulting type of model is termed semi-continuous HMM; its output distributions

$b_j(x) = \sum_{k=1}^{K} b_{j,k} . gk(x)$ with $gk(x) = N(x \mid \mu_k, \sum_k)$

all share the same global set $g = (g_1,...,g_k)$ of Gaussians regardless of the state index j . The semi-continuous model is therefore characterized by the statistics $\lambda = (\pi, A, B, g)$, where density function g_k is represented parametrically by its mean vector μ_k and co-variance matrix $\sum_K$. Evidently, our notation suggests that $\pi, A,$ and B can be interpreted as an ordinary discrete

HMM and g as the codebook of a $K-class$ probabilistic (soft) vector quantizer, transforming continuous feature vectors x into a likelihood array $(g_1(x),...,g_k(x))^T$.

2.4 Support Vector Machines (Svm)

The utilization of support vector machine (SVM) [9], [10] classifiers has gained immense popularity in the last years. SVMs have achieved excellent recognition results in various pattern recognition applications [10]. Also in off-line optical character recognition (OCR) they have been shown to be comparable or even superior to the standard techniques like Bayesian classifiers or multilayer perceptrons [11]. SVMs are discriminative classifiers based on Vapnik's structural risk minimization principle. They can implement flexible decision boundaries in high dimensional feature spaces. The implicit regularization of the classifier's complexity avoids overfitting and mostly this leads to good generalizations. Some further properties are commonly seen as reasons for the success of SVMs in real-world problems: the optimality of the training result is guaranteed, fast training algorithms exist and little a-priori knowledge is required, i.e. only a labeled training set.

Here, we provide a brief introduction to support vector classification. For more details and geometrical interpretations please refer to the standard literature, e.g. by Burges [9] or Cristianini and Shawe-Taylor [10].

Consider a two-class classification problem and a set of training vectors $\{P_i\}_{i=1,...,M}$ with corresponding binary labels $S_i = 1$ for the "positive" and $S_i = -1$ for the "negative" class. In classification an SVM assigns a label S' to a test vector T by evaluating

$$f(T) = \sum_i x_i S_i K(T, P_i) + b \text{ and } S' = sign(f(T)) \quad (1)$$

The weights αi and the bias b are SVM parameters and adopted during training by maximizing

$$L_D = \sum_i x_i - \frac{1}{2}\sum_{i,j} x_i x_j S_i S_j K(P_i, P_j) \quad (2)$$

under the constraints

$$0 \le xi \le C \text{ and } \sum_i x_i S_i = 0 \quad (3)$$

with C a positive constant weighting the influence of training errors. K (·, ·) is the kernel of the SVM. A solution for the αi implies a value for b. The SVM framework gives some flexibility in designing an appropriate kernel for the underlying application. Many implementations of kernels have been proposed so far, one popular example is the Gaussian kernel

$$K(P_i, P_j) = \exp(-r \left\| p_i - p_j \right\|^2) \quad (4)$$

If K (·, ·) is positive definite, (1)–(2) is a convex quadratic optimization problem, for which the convergence towards the global optimum can be guaranteed. However, obtaining this solution for real-world problems can be quite demanding and requires sophisticated optimization algorithms like chunking, decomposition or sequential minimal optimization [10].

Usually αi = 0for the majority of i and thus the summation in (3) is limited to a subset of the Pi, which therefore is called the set of support vectors. Extensions of the binary classification to the multi-class situation are suggested in several approaches [9], [12].

2.6 Radial Bass Function Neural Network (Rbf-Nn)

To improve the accuracy, we have trained RBFNN with the output of both the algorithms. Different samples of Tamil Characters are taken and given as input to both HMM and SVM. If HMM or SVM gives a false character, the neural network is trained with the weightage of both the algorithms and the actual character. This process is done for all the possible false recognition of the two algorithms. During OCR When both the algorithms not giving same character, trained RBFNN is used to retrieve the actual character. This way we can increase the accuracy of OCR.

Radial Basis Functions emerged as a variant of artificial neural network in late 80's.RBF's are embedded in a two layer neural network, where each hidden unit implements a radial activated function [13]. Due to their nonlinear approximation properties, RBF networks are able to model complex mappings, which perception by means of multiple intermediary layers [14].

Radial basis networks can require more neurons than standard feed forward back propagation networks, but often they can be designed in a fraction of the time it takes to train standard feed forward networks. They work best when many training vectors are available. RBF networks have been successfully applied to a large diversity of applications including interpolation [15], image restoration [16], shape-from-shading [17], 3-D object modeling [18], data fusion [19], etc.

3. Experimental Results

We chose "Thirukural" OCR to test the proposed methodologies efficiency. Thirukural [20] of Thiruvalluvar is the most popular, most widely esteemed Tamil Classic of all times. It is a Tamil book on philosophy and life in general, written by Thiruvalluvar, a sage and philosopher,

about 2000 years ago. Its appeal is universal. It is the only Tamil literary work (possibly after the Bible) that has been translated many times in almost all languages of the world. Every "kural" has got 2 lines or 7 words. We tested the efficiency with set of "kural"s. The experimental results are listed in Table 1.

Table 1. Recognition Results for Thirukural

Count of Kurals	*Efficiency*
50	94.1
100	94.3
150	92.5
200	94.2
250	92.3

4. Conclusion

In this paper we have proposed a new method of Tamil OCR where we have fused two algorithms to get the maximum possible efficiency of both the algorithms. Our work primarily deals with choosing a better algorithm, and fusing them finally to attain better accuracy. We have chosen HMM, SVM to be fused and finally used Neural network to predict the correct character when there arises a situation where two algorithms yield two different characters. The computational time of the approach of fusing two algorithms is not discussed in this paper which we may take to our future work. The experimental results show that the accuracy is really improved than the previous works.

References

[1] G. Nagy. Twenty years of document image analysis in pattern analysis and machine intelligence. IEEE Tran. PAMI, pages 38–82, 2000.

[2] Mantas, J., 1986. An overview of character recognition methodologies, Pattern recognition, 19 (6): 425-430.

[3] Govindan, V.K. and A.P. Shivaprasad, 1990. Character Recognition-A Review, Pattern Recognition, 23 (7): 671-683.

[4] Hewavitharana, S, and H.C. Fernando, 2002. A Two Stage Classification Approach to Tamil Handwriting Recognition, pp: 118-124, Tamil Internet 2002, California, USA.

[5] Chinnuswamy, P., and S.G. Krishnamoorthy, 1980. Recognition of Hand printed Tamil Characters, Pattern Recognition, 12: 141-152.

[6] R.M. Suresh, S. Arumugam and K.P.Aravanan, "Recognition of handwritten Tamil characters using fuzzy classifi catory approach ", Proc. The Tamil Internet 2000 Conference, Singapore, July 2000.

[7] Siromoney et al., 1978. Computer Recognition of Printed Tamil Character, Pattern Recognition 10: 243-247.

[8] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Offline Cursive Handwriting Recognition using Hidden Markov Models. Pattern Recognition, 28(9):1399–1413, 1995.

[9] C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121–167, 1998.

[10] N. Cristianini and J. Shawe-Taylor. Support Vector Machines. Cambridge University Press, 2000.

[11] D. DeCoste and B. Schölkopf. Training invariant support vector machines. Machine Learning, 46(1/3):161, 2002.

[12] G. C. Cawley. MATLAB support vector machine toolbox (v0.50β). University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000. URL http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox.

[13] Adrian G. Bors, "Introduction of the Radial Basis Function (RBF) Networks", Department of Computer Science, University of York, UK.

[14] Haykin, S. (1994) Neural Networks: A comprehensive Foundation. Upper Saddle River, NJ; Prentice Hall

[15] Broomhead, D.S., Lowe, D. (1988) "Multivariable functional interpolation and adaptive networks," Complex Systems, vol.2, pp.321-355.

[16] Cha, I., Kassam, S.A., (1996) " RBFN restoration of nonlinearly degraded images," IEEE Trans. on Image Processing, vol. 5, no. 6, pp. 964-975.

[17] Wei, G.-Q., Hirzinger, G., (1997) " Parametric shape -from-shading by radial basis functions, " IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 4, pp. 353-365.

[18] Bors, A.G., Pitas, I., (1999) " Object classification in 3-D images using alpha-trimmed mean radial basis function network," IEEE Trans. on Image Processing, vol. 8, no. 12, pp. 1744-1756.

[19] Chatzis, V., Bors, A. G., Pitas, I., (1999) "Multimodal decision-level fusion for person authentification," IEEE Trans. on Systems, Man, and Cybernetics, part A: Systems and Humans, vol. 29, no.6, pp. 674-680.

[20] "Thirukural of Thiruvalluvar" from www.thirukural.com/

[21] T. Jolliffe, "Principle Component Analysis" Spring-Verlag, New York, 1986.

[22] C. Jain, U.Halici, I. Hayashi, S.B. Lee and S.Tsutsui. "Intelligent biometric techniques in fingerprint and face recognition" 1999, the CRC Press.

[23] Maio and D. Maltoni. "Direct gray-scale minutiae detection in fingerprints" IEEE Trans. Pattern Anal. And Machine Intell. 19(1):27-40, 1997.

[24]"Gabor Filter" from http://en.wikipedia.org/wiki/Gabor_filter

[25] L.R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, vol. 77 no. 2, 1989, pp. 257-286.

[26] Y.He, A.Kundu: 2-D shape classification Using Hidden Markov Model, IEEE Trans. On PAMI, vol.13,1991, pp.1172-1184.

[27] F.Samaria, F.Fallside: Face Identification and Feature Extraction Using Hidden Markov Models, in G.Vernazza, A.N.Venetsanopoulos, C.Braccini (editors): Image Processing: Theory and Applications, Elsevier Science publishers B.V., 1993, pp.292-302.

[28] A.Kundu, Y.He, P.Bahl: Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach, Pattern Recognition, 22(3), 1989, pp.283-297.

[29] G. Nagy, On the Frontiers of OCR, Proceedings of the IEEE, vol. 40, #8, pp. 1093-1100, July 1992.

R. Jagadeesh Kannan received BE Instrumentation & Control Engineering from Madurai Kamaraj University in 2000. He received his ME Computer Science & Engineering from Manonmaniam Sundaranar University in 2002. Currently he is pursuing his research work in the field of Pattern Recognition. He is now with the Department of Computer Science & Engineering, RMK Engineering College, Kavaraipettai, Chennai, India. His interests are in Pattern Recognition, Artificial Intelligence & Image Processing. He is currently a member of Computer Society of India (CSI) & Indian Society for Technical Education (ISTE).