2011 6th International Conference on Industrial and Information Systems, ICIIS 2011, Aug. 16-19, 2011, Sri Lanka

# A Novel SIFT-based Codebook Generation for Handwritten Tamil Character Recognition

A. Subashini and N. D. Kodikara

***Abstract*—A method for the off-line recognition of Tamil handwriting characters based on local feature extraction is investigated. In the proposed method each pre-processed character is represented by a set of local SIFT feature vectors. From a large set of SIFT descriptors, the key idea is to create a codebook for each character using *K*-means clustering algorithm. *K*-means is an optimisation algorithm but this algorithm takes very long time to converge. We construct an initial codebook by using the Linde Buzo and Gray (LBG) algorithm so that the convergence time for *K*-means is reduced considerably. Target character is recognised into one of twenty categories by *k*-nearest neighbour classification. An average recognition rate of 87% on the character level has been achieved in experiments using six thousand training and two thousand testing images of twenty selected characters. Further study may include more characters and more samples being recognised with better classifier.**

***Index Terms*—Character Recognition, *K*-means, *k*-NN, SIFT, Tamil Handwritten Characters.**

## I. Introduction

As the developments in the computer field are tremendous, there is a need to improve the man-machine interface. If computers can be made intelligent enough to understand human handwritings, it will be possible to make computer interfaces more ergonomic and attractive. Recognition of any handwritten characters with respect to any language is difficult, since, the handwritten characters differ in written format, intensity, scale and orientation, not only from person to person but also according to the state of mood of the same person.

As a result of intensive research and development efforts, character recognition systems are available for English [11], Chinese [14], and Japanese languages. Some efforts have been reported in the literature for Tamil scripts [1], [4], [6], [7], [13], [15], [18]. Tamil character recognition (TCR) is still far from the final frontier, this is a result of the lack of utilities such as Tamil text databases, dictionaries etc. The problem of TCR is more difficult than other languages in respects to the similarity and complexity of characters that are composed of circles, holes, loops, and curves. Hence, Tamil handwriting recognition requires more research to reach the ultimate goal of machine simulation of human reading.

In recent times, Tamil is being extensively used in computers by international Tamil community. As Tamil is official and spoken language in several countries, the use of Tamil in Information Technology will be more in future. In order to promote this further, we have developed a system to recognise the Handwritten Tamil Characters, which may be useful for recognising Tamil texts.

Selection of a feature extraction method is probably the most important factor in achieving high recognition performance in character recognition systems [17]. Although many feature extraction approaches have been proposed in the field of handwritten character recognition, most of them have different kinds of weakness while being used in a practical system. Furthermore, a recognition system which is suitable for one language need not be apt for another. To our best knowledge in the literature of Tamil character recognition, till date, no work has been reported with the use of bag-of-feature representation. Many existing research works are based on using zone/grid features [4], [13], geometrical features [15], [18] and chain code histogram features [1].

Recently, a very powerful feature extraction method called Scale Invariant Feature Transform (SIFT), has been proposed by D. G. Lowe [10]. SIFT comprises keypoint localisation and construction of keypoint descriptor. SIFT, especially the SIFT descriptor, has been widely employed for many image applications and proven to be very effective such as object recognition in computer vision [2], [12], texture classification [21], and image classification in cell biology [8].

Most recently, a modified version of SIFT features are being used for character recognition, but most of the work was confined to handwritten Chinese character recognition with good performance [19], [20], [22]. In this paper, we have made an attempt to recognise handwritten Tamil characters by using SIFT features .We have presented a new and efficient character recognition based on bag-of-keypoints representation. This bag-of-keypoints method is based on vector quantization of SIFT descriptors of image patches. The *K*-means clustering algorithm has been used to generate a codebook for every character. In addition, an analysis has been carried out to determine the size of the codebook to achieve high performance of clustering in the recognition.

The rest of this paper is organised as follows. Section 2 presents a brief description of Tamil script and the present database of Tamil handwritten isolated characters. Section 3 describes the proposed system in detail. Results of our experimentation are presented in section 4. Finally, section 5, concludes our work with possible extensions.

## II. Handwritten Database for Character Set

### A. Tamil Script

Tamil, a member of the Dravidian language family, is spoken by around 72 million people in the world. The alphabets of Tamil are very old and are organised in a systematic way.

A. Subashini is with the Computer Unit, University of Jaffna, Sri Lanka. (e-mail: subashini@jfn.ac.lk).

N. D. Kodikara is with the Department of Information Systems Engineering, University of Colombo School of Computing (UCSC), Sri Lanka. (e-mail: ndk@ucsc.cmb.ac.lk).



978-1-61284-0035-4/11/$26.00 ©2011 IEEE

2011 6th International Conference on Industrial and Information Systems, ICIIS 2011, Aug. 16-19, 2011, Sri Lanka

அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ ஔ
க ங ச ஞ ட ண த ந ப ம ய ர ல வ ழ ள ற ன
ஸ்ரீ ஜ ஷ ஸ ஹ க்ஷ
ஃ

Fig. 1. Basic Tamil characters

The alphabet set splits into set of twelve vowels, eighteen consonants. The modern script of Tamil also consists of another five consonants. Thus, this script has 35 basic characters. Also, there is another character, called aytam which is classified in Tamil grammar as being neither a consonant nor a vowel. In addition to the above isolated characters, often a consonant or a cluster of two or more consonants combines with a vowel causing a modified shape of the vowel, called vowel diacritic (12×18). Although this indicates the presence of a large number of Tamil characters (more than 247), the modern Tamil language does not use many of these combinations and only 156 characters including independent vowels, consonants and their combinations are presently used for writing in Tamil. Fig. 1 shows the basic Tamil characters.

### B. Data Collection

The dataset hpl-tamil-iso-char-offline-1.0, developed by HP Lab India [5], is used for the present work. We have considered 20 characters of the Tamil alphabets for our study.

## III. Proposed System

The main steps of our method are:

- Pre-processing the character images.
- Detection of interest points in character images.
- Constructing visual codebooks by means of clustering techniques (*K*-means). The codebook is the set of centres of the learnt clusters.
- Constructing a bag-of-keypoints, which counts the number of patches assigned to each cluster.
- Applying a *k*-NN classifier, treating the bag-of-keypoints as the feature vector, and thus determine which character to assign to the image.
- Selection of a codebook that yields the best overall classification accuracy.

The methodology consists of four major sections: Pre-processing, feature extraction, codebook generation, and classification.

### A. Pre-processing

Pre-processing covers all those functions carried out prior to feature extraction to produce a cleaned up version of the original image so that it can be used directly and efficiently by the feature extraction components of the TCR.

The steps in pre-processing involves:

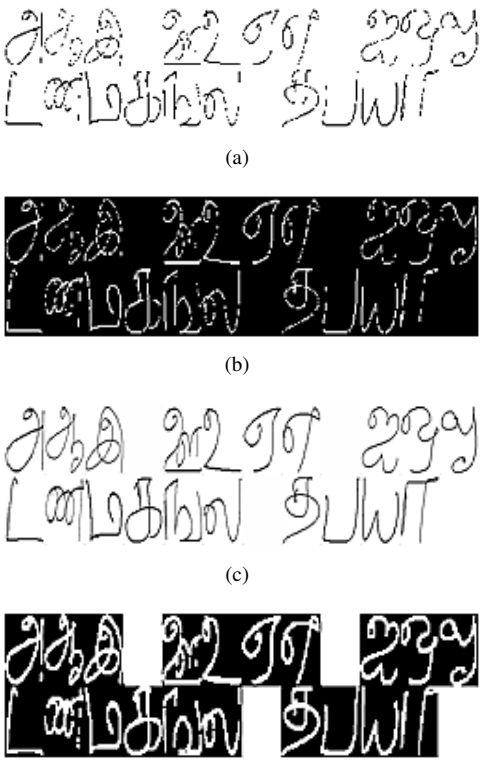
- Noise reduction: Is achieved by morphological operations [3].
- Size normalization: Bicubic interpolation is used for standard sized image.
- Extraction of interest region: Is done using morphologic gradient to find character boundaries.
- Segmentation: Since in Tamil, the characters are always written in "print fashion", not connected, histogram profile and connected component analysis are able to handle the line and character segmentation problem.

Fig. 2 shows sample, binary, smoothed and gradient images of an input.

### B. Feature Extraction

The SIFT algorithm takes a character image and transforms it into a set of local features, each of which describes a local part around a keypoint. Each of these feature vectors is supposed to be distinctive and invariant to any scaling, rotation or translation of the image. The SIFT descriptor builds a representation for each keypoint based on a patch of pixels in its local neighbourhood. For each sample character, the 128-dimensional SIFT features are calculated. Derived all SIFT features of a particular character are concatenated to make bigger feature space. In Fig. 3, keypoints are indicated as arrows of a Tamil character.

Different sized images were used in the study ranging from 32×32 pixels to 128×128 pixels. When the image size was small, it captured less SIFT features. However, due to the varying nature of handwriting, there was high dissimilarity between the feature vectors of the same class. Large sized



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

Fig. 2. Pre-processed image of an Input image. (a) Original Image, (b) Binary Image, (c) Smoothed Image, and (d) Gradient Image

2011 6th International Conference on Industrial and Information Systems, ICIIS 2011, Aug. 16-19, 2011, Sri Lanka

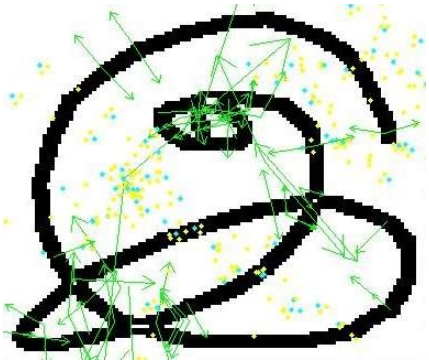


Fig. 3. An image of a Tamil character shows its own SIFT features

image failed to capture the essential patch of characters, which make them distinct from others. Since the best results were produced by 48×48 sized images, we decided to use 48×48 sized images for feature extraction.

### C. Codebook Generation

Vector Quantisation (VQ) can be defined as a mapping function that maps 128-dimensional vector space to a finite set CB = $\{C_1, C_2, C_3, \cdots, C_N\}$. The set CB is called a codebook consisting of N number of codewords and each codeword $C_i$ is of dimension 128. The key to VQ is the good codebook. The collections of features are fed to *K*-means clustering algorithm to create a codebook. Then the output of clustering is dispatched to the decision making process. The *K*-means algorithm divides the set of training feature vectors into *K* disjoint clusters in such a way that the following two necessary conditions for optimality are satisfied.

- Nearest Neighbour Condition: The optimal quantiser is realised by using a minimum distortion.
- Centroid Condition: Each vector is chosen to minimize the average distortion in a cluster. Such a vector is called the centroid of that cluster which is equivalent to the statistical mean of the cluster members.

The quantised codeword is selected to be the closest in Euclidean distance from the input feature vector.

Select *K* random keypoints from the training space and call it as an initial codebook. Find the squared Euclidean distance of all the training vectors with the selected *K* codewords and *K* clusters are formed. A training keypoint $T_j$ is put in ith cluster if the squared Euclidean distance of $T_j$ with ith codeword is minimum. Centroids of each of cluster form set of new codebook as an input to *K*-means algorithm for the next iterations. Compute mean squared error (MSE) for each of *K* clusters and net MSE.

The optimal solution of *K*-means algorithm depends on the random initial selection of the codebook [16]. This initial selection is usually far off from the optimal solution. Hence it takes extremely huge time to converge. There is very low probability that the initial solution is close to the optimal solution. In this paper we are proposing *K*-means algorithm for optimisation of codebook which already exists. For demonstration we have used codebooks obtained from LBG algorithm [9]. It is observed that this algorithm converges faster by reducing the convergence time by a factor of more than three.

## IV. Classification

Finally, a *k*-Nearest Neighbour (*k*-NN) classifier is used for the recognition process. The bag-of-keypoints are used as feature vectors for classification. The classification accuracy is influenced by the number of *k* nearest neighbours. We tried multiple values for *k*, as result, the highest accuracy was mostly given by *k* = 1. Histogram equalisation is applied to get the final result. The overall architecture of the Tamil character recognition system developed in this study is shown below in Figure 4:

## V. Results and Discussion

The proposed approach is implemented on a desktop PC running with Intel processor 2.16 GHz and 2GB RAM. The implementation is written in MATLAB due to its simplicity. We trained the system with 6000 character images belonging to selected twenty classes. The testing data contained a separate set of 2000 characters. In the experiment, the size of the codebook was varied from 128, 256, 512 to 1024, and 1024 sized codebook had higher recognition rate when compared to others. Table 1 shows recognition results against the codebook size. Figure 4 shows error variation vs. number of character of *k*-NN classification for codebook size 1024. The overall recognition rate realised as 87%.

## VI. Conclusions and Recommendations

This paper presents a bag-of-keypoints approach to offline Tamil handwritten character recognition. A bag-of-keypoints corresponds to histogram of number of occurrences of particular image patterns in a given image. We have presented a simple but novel approach to character recognition using SIFT feature vectors constructed from character image patches. This approach has been evaluated on a twenty different Tamil characters image database. The main advantages of the method are its simplicity and computational efficiency.

Our testing result shows that the algorithm works well for the selected set of 20 characters. The algorithm is tried

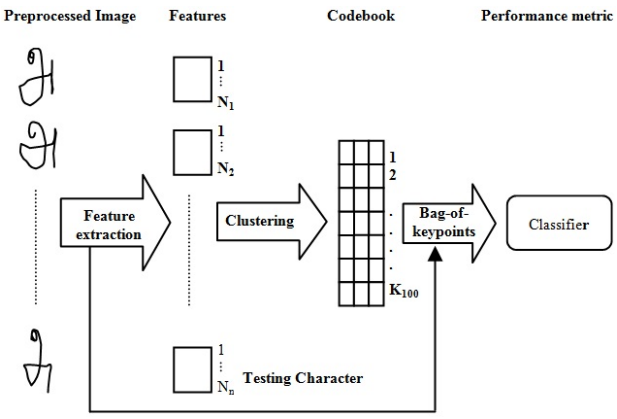



Fig. 4. Architecture of the Proposed System for Tamil Character recognition

2011 6th International Conference on Industrial and Information Systems, ICIIS 2011, Aug. 16-19, 2011, Sri Lanka

TABLE I
RECOGNITION RESULTS AGAINST VARIOUS CODEBOOK SIZES. THE VALUES INDICATE THE NUMBER OF CORRECTLY CLASSIFIED CHARACTERS EACH OUT OF 100.

| Char | அ | ஆ | இ | ஈ | உ | ஊ | எ | ஏ | ஐ | ஒ | ஓ | ம | க | ங | ல | ட | ண | த | ப | ய |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CB 1024 | 96 | 95 | 99 | 86 | 72 | 88 | 68 | 91 | 99 | 67 | 79 | 64 | 98 | 99 | 97 | 79 | 99 | 93 | 75 | 98 |
| CB 512 | 89 | 89 | 98 | 74 | 58 | 77 | 66 | 83 | 99 | 66 | 59 | 58 | 96 | 99 | 85 | 78 | 96 | 86 | 73 | 96 |
| CB 256 | 90 | 86 | 99 | 71 | 59 | 69 | 58 | 85 | 94 | 61 | 60 | 58 | 92 | 97 | 89 | 75 | 96 | 85 | 62 | 91 |

for different standard image sizes of 32×32, 48×48, 64×64, 96×96 and 128×128.

The overall recognition rate varies from 64 to 99% for codebook size 1024. The experimentation result shows that the *k*-NN based approach for 48×48 sized images recognises well when compared to other sizes. We show that, in practice, the codebook generation approach produces near optimal accuracy rate and better time performance. The main recognition errors were due to abnormal writing and ambiguity among similar shaped characters.

Our testing results indicate that the proposed method is to be extended by considering the entire character classes as well as using a large database of handwritten character samples. The approach can be extended to recognise other language scripts as well. In future, we will be directed toward using local features to recognise Tamil words. Also, when considering all the Tamil characters with the proposed system, the traditional *K*-means method would find it difficult to cope with large scale data. Therefore we need some fast approaches in constructing visual codebooks such as discussed in [12]. Furthermore, we would like to employ support vector machine (SVM) classifier which has better performance in higher dimensional spaces.

## REFERENCES


[1] Bhattacharya, U., Ghosh, S. K., and Parui, S.K., *A Two Stage Recognition Scheme for Handwritten Tamil Characters*, Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[2] Csurka, G., Dance, C.R., Fan, L., Willamowski, J., and Bray, C., *Visual categorization with bags of keypoints*, In ECCV Workshop on Statistical Learning in Computer Vision, 2004.

[3] Gonzalez, R.C. and Woods, R.E., *Digital Image Processing using MATLAB*, Addison Wesley publishers, 1993.

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

[5] HP Labs Isolated Handwritten Tamil Character Dataset, http://www.hpl.hp.com/india/research/penhw-interfaces-linguistics.html

[6] Indra Gandhi R. and Iyakutti, K., *An Attempt to Recognized Handwritten Tamil, Character Using Kohonen SOM*, International Journal of Advanced Networking and Applications, Vol. 01 Issue: 03 pp. 188–192, 2009.

[7] Jagadeeshkannan, R. and Prabhkar, R., *Off-Line Cursive Handwritten Tamil Character Recognition*, WSEAS Transactions on Signal Processing, issue 6, Vol. 4, 2008.

[8] Ji, S., Li, Y-X., Zhou, Z-H., Kumar, S., and Ye, J., *A bag-of-words approach for Drosophila gene expression pattern annotation*, BMC Bioinformatics, 2009.

[9] Linde, Y., Buzo, A., and Gray, R.M., *An algorithm for vector quantizer design*, IEEE Trans. Commun., Vol. COM-28, No. 1, pp. 84–95, 1980.

[10] Lowe, D. *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60, 2, pp. 91–110, 2004.

[11] Plamondan, R. and Srihari, S.N., *Online and off-line Handwriting Recognition: A Comprehensive Survey*, IEEE Trans on PAMI, Vol. 22, No 1, pp 63–84, 2000.

[12] Ramanan, A. and Niranjan, M., *A One-Pass Resource-Allocating Codebook for Patch-based Visual Object Recognition*, In proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 35–40, 2010.

[13] Shanthi, N. and Duraiswamy K., *A novel SVM-based handwritten Tamil character recognition system*, Pattern Anal Applications, 13: pp. 173–180, 2010.

[14] Srihari, S.N., Yang, X., and Ball, G.R. *Offline Chinese handwriting recognition: an assessment of current technology*, Front. Comput. Sci. China, 1(2): 137–155, 2007.

[15] Sureshkumar, C. and Ravichandran, T., *Character Recognition using RCS with Neural Network*, International Journal of Computer Science Issues (IJCSI), Vol. 7, Issue 5, September 2010.

[16] Theodordis, S. and Koutroumbas, K., *Pattern Recognition*, Fourth edition, Academic Press an imprint of Elsevier, 2009.

[17] Trier, O.D. and Jain, A.K., *Feature extraction methods for character recognition - a survey*, Pattern Recognition, Vol. 9, No 4, pp. 641–662, 1996.

[18] Venkatesh. J. and Sureshkumar.C., *Handwritten Tamil Character Recognition Using SVM*, International Journal of Computer and Network Security (IJCNS), Vol. 1, No. 3, December 2009.

[19] Wu, T. Kai-yue, Q. et al., *An Improved Descriptor for Chinese Character Recognition*, Third IEEE International Symposium on Intelligent Technology Application, 2009.

[20] Zhang, Z., Jin, L., Ding, K., and Gao, X., *Character-SIFT: A Novel Feature for Offline Handwritten Chinese Character Recognition*, In Proceedings of ICDAR'2009. pp. 763–767, 2009.

[21] Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C., *Local features and kernels for classification of texture and object categories: A comprehensive study*, CVPRW 2006.

[22] Zhen, J., Kai-yue, Q., and Kai, C., *SSIFT: An Improved SIFT Descriptor for Chinese Character Recognition in Complex Images*, Computer Network and Multimedia Technology, 2009.