Handwritten Devanagari Word Recognition: A Curvelet Transform Based Approach

Brijmohan Singh
Department of CS&E
College of Engineering Roorkee
Roorkee-247667, India

Ankush Mittal
Department of CS&E
College of Engineering Roorkee
Roorkee-247667, India

M.A. Ansari
Department of Electrical Engineering
Galgotias College of Engineering and Technology
Gr. Noida, India

Debashis Ghosh
Department of E&C
IIT Roorkee
Roorkee-247667, India

Abstract— **This paper presents a new offline handwritten Devanagari word recognition system. Though Devanagari is the script for Hindi, which is the official language of India, its character and word recognition pose great challenges due to large variety of symbols and their proximity in appearance. In order to extract features which can distinguish similar appearing words, we employ Curvelet Transform. The resultant large dimensional feature space is handled by careful application of Principal Component Analysis (PCA). The Support Vector Machine (SVM) and k-NN classifiers were used with one-against-rest class model. Results of Curvelet feature extractor and classifiers have shown that Curvelet with k-NN gave overall better results than the SVM classifier and shown highest results (93.21%) accuracy on a Devanagari handwritten words set.**

***Keywords*- OCR; Devanagari; Curvelet Transform; SVM; k-NN**

I. INTRODUCTION

Many of the challenges in Optical Character Recognition (OCR) research are raised in the area of handwritten character and word recognition. Real-world handwriting is a mixture of cursive and noncursive parts, which makes the problem of recognition significantly difficult. Offline handwritten word recognition is an important area of Document Analysis and Recognition (DAR). DAR is a mechanism in which the document images are processed to obtain text and graphics features. The main objective of DAR is to read the intended information from the document using computer as much as a human would do. The outcome of DAR system is usually in the ASCII format [1]. The applications of DAR [2-3] include such as office and library automation, publishing houses, help to the visually handicapped when interfaced with a voice synthesizer, postal service assistance, reading entrance examination forms, processing of applications of victims and criminal records in police station, etc. A slight mistake in interpreting the characters can lead to mistake in the automation process such as wrong dispatch in postal service or wrong entry in entrance examination forms.

Handwriting recognition can be achieved by character, word and sentence level. A character recognizer needs to be trained with sample characters from the alphabets used in the language. There are two approaches for the recognition of isolated handwritten Devanagari words [4]. The first is to segment the word into its character parts, individually recognize each character, and then reconstruct the word. The major drawback of this approach for the Devanagari script is that the words contain Matra, Shirorekha, conjunct characters, modifiers and lack of standard benchmark database for training the classifier. The second scheme is to recognize the word in its entirety. Word recognizers are complex if they are general purpose but are simpler if it is based on specific lexicon. This approach of word recognition avoids the overhead of character segmentation.

While significant advances have been achieved in recognizing Roman based scripts like English, ideographic characters (Chinese, Japanese, Korean, etc) and Arabic to some extent, OCR research on Indian scripts is very less. Only few works on some of the major scripts like Devanagari, Bangla, Gurumukhi, Tamil, Telgu, etc. are available in the literature.

The era of handwritten Devanagari character recognition was started in the early days of OCR research by Sethi et al. [5]. The research in offline Devanagri word recognition was started by Parui et al. proposed a HMM based holistic approach for the word recognition [4]. Later, Shaw et al. published a segmentation based approach [6].

Recently, a Curvelet-based SVM recognizer has been proposed in [7] for recognition of handwritten Bangla characters with an overall accuracy of 95.5%. Since, Devanagari and Bangla belong to the same Brahmic family of scripts having a common origin; many similarities are observed among their characters. Consequently, their characteristic features are somewhat close to each other and hence, many character recognition algorithms are expected to be equally applicable to Devanagari, Bangla and other scripts belonging to the Brahmic family. In view of this, we propose a Curvelet based feature extractor with SVM and k-NN classifiers for offline handwritten Devanagari word recognition system.

In our present work for word recognition, we have applied the holistic approach to avoid the overhead of segmentation and due to lack of standard benchmark database for training the classifier. Since a standard benchmark database was not available for Indian script so we created a word database for Devanagari to test the performance of our system. In the present report, training and test results of the proposed approach are presented on the basis of this database.

II. Features of Devanagari Script

Devanagari is the script used for writing Hindi which is the official language of India [8]. It is also the script for Sanskrit, Marathi and Nepali languages. Devanagari script consists of 13 vowels and 33 consonants characters. These characters are called the basic characters. The characters may also have a half form. A half character in most of the cases touches the following character, resulting in a composite character. Some characters of Devanagari script take the next character in their shadow because of their shape. The script has a set of modifier symbols which are placed either on top, at the bottom, on the left, to the right or a combination of these. Top modifiers are placed above the shirorekha (Head line), which is a horizontal line drawn on the top of the word. The lower modifiers are placed below the character which may or may not touch the characters. More than one lower modifier may also be placed below one character. A character may be in shadow of another character, either due to a lower modifier or due the shapes of two adjacent characters. Upper and lower modifiers with basic character modifiers make OCR with Devanagari script very challenging. OCR is further complicated by compound characters that make character separation and identification very difficult.

III. Steps Involved in Word Recognition

This paper attempts to presents a method, which is based on the following important steps: Pre-processing, Curvelet based feature extraction and classification by SVM and k-NN. Figure 1 shows the architecture of proposed system.

3.1 Pre-processing

In the off-line OCR, handwritten image to be recognized is captured by a sensor, for example, a scanner or a camera. Pre-processing of grayscale source image is essential for the elimination of noisy areas, smoothing of background texture as well as contrast enhancement between background and text areas. For smoothing, the input gray level image is first filtered by the Wiener filter [9] and then binarized by the Otsu’s method [10]. The Wiener filtered grayscale image ‘I’ is obtained from source grayscale image ‘Is’ according to formula:

$$I(x,y) = \mu + \frac{(\sigma^2 - v^2)(I_s(x,y) - \mu)}{\sigma^2} \qquad (1)$$

Where μ is the local mean, σ^2 is the variance at 3×3 neighbourhood around each pixel and v^2 is the average of all estimated variance for each pixel in the neighbourhood.

3.2 Feature Extractions

After pre-processing the images, features relevant to the classification are extracted from the smoothed images. The extracted features are organized in a database, which is the input for the recognition phase of the classifier. A feature extraction scheme based on digital Curvelet transform [11] has been used. In this work, the words from the sample images are extracted using conventional methods. A usual feature of handwritten text is the orientation of text written by the writer. Each sample is cropped to edges and resized to a standard width and height suitable for digital Curvelet transform. The digital Curvelet transform at a single scale is applied to each of the samples to obtain Curvelet coefficients as features. In our case, we obtained 1024 (32 x 32) feature coefficients, depending on the size of the input sample image.

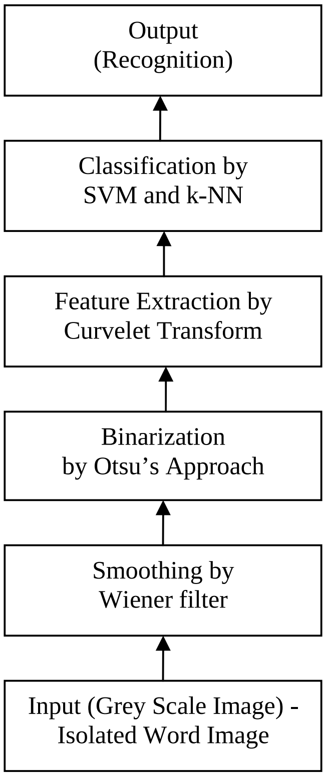


Figure 1: An architecture of proposed OCR

3.2.1 Curvelet Transform

Word recognition earlier was handled by string edit distance [4] and scalar features [6]. However, for large set of characters, as in Devanagari language, automatic curve matching is highly useful. Considering this, we explored the use of curvelet transform which represents edges and singularities along curves more precisely with the needle-shaped basis elements. The elements own super directional sensitivity and smooth contours capturing efficiency. Since Curvelets are two dimensional waveforms that provide a new architecture for multiscale analysis, they can be used to distinguish similar appearing characters better.

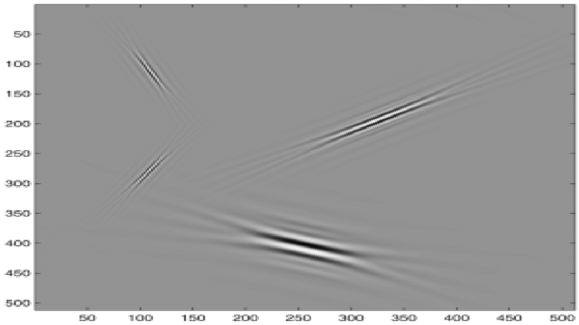


Figure 2: Elements of Curvelet

The Curvelet frame preserves the important properties, such as parabolic scaling, tightness and sparse representation for surface-like singularities of co-dimension one. Figure 2 shows a sample of digital Curvelet. Since many of the characters in a word not only consist of edge discontinuities but also of curve discontinuities. The most widely used Wavelet transform works well with edge discontinuities but a curve discontinuity affects all the Wavelet coefficients. On the other hand, the curve discontinuities in any character or word are well handled with Curvelet transform with very few numbers of coefficients. Hence, Curvelet-based feature are likely to work well for Devanagari character and word recognition.

The curvelet transform includes four stages:

1. Sub-band decomposition
2. Smooth partitioning
3. Renormalization
4. Ridgelet analysis

1. Sub-band decomposition is to divide the image into resolution layers where each layer contains details of different frequencies i.e.,

$$f \mapsto (P_0 f, \Delta_1 f, \Delta_2 f, \ldots) \qquad (2)$$

Here, P_0 is the low pass filter and Δ_1, Δ_2 … are high pass (band pass) filter.

2. Smooth Partitioning: Each subband is smoothly windowed into “squares” of an appropriate scale i.e.

$$\Delta_s f \rightarrow (\omega_Q \Delta_S f)_{Q \in Q_S} \qquad (3)$$

Where, Q_s denotes the dyadic square of side 2^{-s} and ω be a smooth windowing function with ‘main’ support of size $2^{-s} \times 2^{-s}$.

3. Renormalization: Each resulting square is renormalized to unit square i.e.

$$g_Q = 2^{-s}(T_Q)^{-1}(\omega_Q \Delta_S f), Q \in Q_S \qquad (4)$$

4. Ridgelet analysis: Each square is analyzed in the ortho-ridgelet system i.e.

$$\alpha_\mu = \langle g_Q, p_\lambda \rangle, \mu = (Q, \lambda) \qquad (5)$$

3.3.2 Dimensionality Reduction

A significant problem in using Curvelet transform is that it gives a large dimensional feature space. Any classifier will require a lot of training data when the feature space is large as well as it will be time consuming. Dimensionality reduction is therefore an obvious choice.

There are several methods of dimensionality reduction. Some methods such as [12] [13], select a few prominent features out of all the features. Others like PCA transform the feature space into a reduced set of features preserving the information as far as possible [14]. Since Curvelet is a mathematical tool which

generates features. PCA is a natural choice. PCA provides a way to identify "patterns" in data and expressing the data in order to highlight the correlations such as similarities and dissimilarities.
The first few eigen values from PCA will contain most amount of information in the present problem, which does not contain dense information. Thus, we chose to use 200 numbers of eigen values for PCA from original 1024 features. These features covered 95% variance in feature space.

3.4 Classification

The main task of classification is to use the feature vector provided by the feature extraction algorithms to assign the object to a category. A more general task is to determine the probability for each of the possible categories. The abstraction provided by the feature extractor representation of the input data enables the development of a largely domain-independent theory of classification. The degree of difficulty of the classification problem depends on the variability in the feature values for object in the same category relative to the difference between feature values for objects in different categories. The variability of feature values for object in the same category may be due to complexity, and may be due to noise [15].

3.4.1 SVM

Support vector machines (SVM) was developed by Vapnik in 1995 [16] and it is an extensively used tool for pattern recognition due to its many attractive features and promising empirical performance specially in classification and nonlinear function estimation. SVM are used for time series prediction and compared to radial basis function network. The classification problem can be restricted to consideration of the two-class problem without loss of generality.
Consider an example of linearly separable classes. We assume that we have a data set

$$D = \{(x_i, y_i)\}_{\rho=1}^{l} \qquad (6)$$

of labeled example,
where $x \in \Re^n, y_i \in \{-1,1\}$, with a hyperplane, $\langle w, x \rangle + b = 0$,

and we wish to select, among the infinite number of linear classifiers that separate the data, one that minimizes the generalization error, or at least an upper bound in it. Hyperplane with generalization property is the one that leaves the maximum margin between the two classes where, margin is defined as the sum of the distances of the hyperplane form the closest point of two classes.
If vector's set is separated without error and the distance between the closest vectors to the hyperplane is maximum then it is said to be optimally separated. There exists some redundancy in above equation, and without loss of generality.
It is best to consider canonical hyperplane, where the parameters w, b, are considered by,

$$\min\left|\langle w, x^i \rangle + b\right| = 1. \qquad (7)$$

A separating hyperplane in canonical form must satisfy the following constraints,

$$y^i \left[\langle w, x^i \rangle + b\right] \geq 1, i = 1,......., l. \qquad (8)$$

The distance d (w,b,x) of a point x from the hyperplane (w,b) is,

$$d(w,b,x) = \frac{\left|\langle w, x^i \rangle + b\right|}{\|w\|}. \qquad (9)$$

Hence, the hyperplane that optimally separates the data is one that minimizes

$$\Phi(w) = \frac{1}{2}\|w\|^2 \qquad (10)$$

If the two classes are non-sharable the SVM looks for the hyperplane that maximizes the margin and that, at the same time minimizes the quantity proportional to the number of misclassification error.

The performance of SVM classification is based on the choice of kernel function and the penalty parameter C. In this work, we used RFB kernel that maps nonlinearly samples into a higher dimensional space, and can handle the case when the relation between class labels and attributes is nonlinear. The RBF kernel can be described as

$$k\left(x,z\right)=\exp\left(-\gamma\times\left\|x-z\right\|^{2}\right) \qquad (11)$$

Thus, while using the RFB kernel functions; there are two parameters C and γ that need to be selected. Usually these parameters are selected on a trial or error basis. In our experiment, we used SVM classifier with Radial Basis Kernel for classification as it has given best results for our dataset. To obtain a more accurate model, the cost factor C of SVM was adjusted. In our case, cost factor C=20, gave the most desirable results. In order to keep the model simple, the cost factor was not further increased.

3.4.2 k-Nearest Neighbour

The k-Nearest Neighbor (k-NN) classifies an unknown sample based on the known classification of its neighbours [18]. Let us suppose that a set of samples with known classification is available, the so-called training set. Intuitively, each sample should be classified similarly to its surrounding samples. Therefore, if the classification of a sample is unknown, then it could be predicted by considering the classification of its nearest neighbor samples. Given an unknown sample and a training set, all the distances between the unknown sample and all the samples in the training set can be computed. The distance with the smallest value corresponds to the sample in the training set closest to the unknown sample. Therefore, the unknown sample may be classified based on the classification of this nearest neighbor. k- NN is an instance-based learning type classifier, or lazy-learning where the function is only approximated locally and all computation is deferred until classification. The training samples are mapped into multidimensional feature space. The space is partitioned into regions by class labels of the training samples. A point in the space is assigned to the class c if it is the most frequent class label among the k nearest training samples. Usually Euclidean distance is used. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the actual classification phase, the same features as before are computed for the test sample (whose class is not known). Distances from the new vector to all stored vectors are computed and k closest samples are selected. The new point is predicted to belong to the most numerous classes within the set. The best choice of k depends upon the samples; generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by parameter optimization using, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbour algorithm. The accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the features scales are not consistent with their relevance.

When given an unknown data, the k-nearest neighbour classifier searches the pattern space for the k training data that are closest to the unknown data. These k training tuples are the k “nearest neighbours” of the unknown data. “Closeness” is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say, X_1 = (x_{11}, x_{12},…, x_{1n}) and X_2 = (x_{21}, x_{22},... , x_{2n}), is

$$dist(X_{1},X_{2})=\sqrt{\sum_{i=1}^{n}(X_{1i}-X_{2i})} \qquad (12)$$

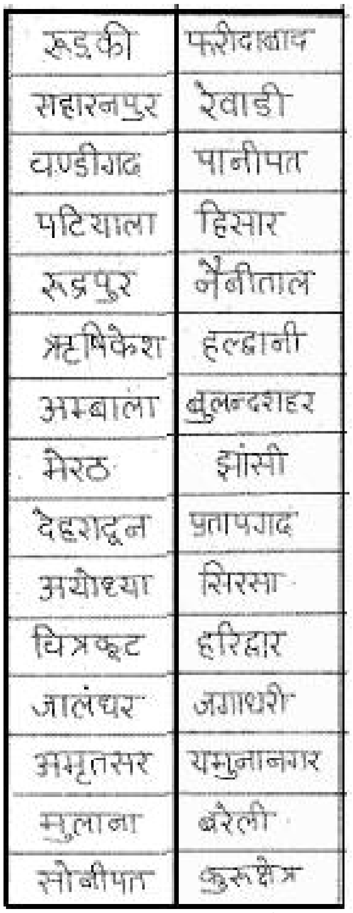
Typically, we normalize the values of each attribute. This helps prevent attributes with initially large ranges (such as income) from outweighing attributes with initially smaller ranges (such as binary attributes). Min-max normalization, for example, can be used to transform a value v of a numeric attribute A to v in the range [0, 1] by computing

$$v'=\frac{v-\min_{A}}{\max_{A}-\min_{A}} \qquad (13)$$

IV. EXPERIMENTAL RESULTS

In our experiment, we collected a dataset of 28, 500 handwritten Devanagari words of 30 classes collected from 950 different writers (sample image is shown in figure 2). Feature extraction was done on each sample using Curvelet Transform at a single scale. The feature vector thus obtained (the coefficients) had a dimensionality of 1024. Principal component analysis of the coefficients was done to reduce the size of feature vector to about 200 dimensions. The ratio between training and testing samples was maintained at 75:25 respectively.

The Support Vector Machine (SVM) and k-NN classifiers were used with one-against-rest class model. Experimental results of Curvelet feature extractor with SVM and k-NN classifiers have shown that Curvelet with k-NN gave overall better results than the SVM classifier and shown highest results (93.21%) accuracy on a Devanagari handwritten word set. Table 1 shows the comparison of results of Curvelet transform with SVM and k-NN classifiers.



रूड़की	फरीदाबाद
सहारनपुर	रेवाडी
चण्डीगढ	पानीपत
पटियाला	हिसार
रूद्रपुर	नैनीताल
ऋषिकेश	हल्द्वानी
अम्बाला	बुलन्दशहर
मेरठ	झांसी
देहरादून	प्रतापगढ
अयोध्या	सिरसा
चित्रकूट	हरिद्वार
जालंधर	जगाधरी
अमृतसर	यमुनानगर
मुलाना	बरेली
सोनीपत	कुरूक्षेत्र

Figure 2 Sample image of Dataset

Table 1: Shows the comparison of feature extractor with classifiers

Feature Extractor	**Classifiers (75:25 Training Test Split)**			
	k- NN (%)		**SVM (%)**	
	Accuracy	**Error Rate**	**Accuracy**	**Error Rate**
Curvelet Features	93.21	6.79	85.6	14.4

V. CONCLUSION

This paper describes a holistic system of offline handwritten Devanagari word recognition. In this paper, we proposed a Curvelet feature extractor with SVM and k-NN classifiers based scheme for the recognition of handwritten Devanagari words. The Support Vector Machine (SVM) and k-NN classifiers were used with one-against-rest class model. Results of Curvelet feature extractor with SVM and k-NN classifiers have shown that Curvelet with k-NN gave overall better results than the SVM classifier.

The proposed scheme was tested only on 28,500 samples of 30 Indian city names. However, the accuracy of proposed scheme may be enhanced by increasing the number of training samples and/ or applying the proposed scheme at different resolution scheme. Hence, Curvelet Transform proves to be useful in Devanagari word recognition.

FUTURE SCOPE

Most of the works reported on Indian languages are on good-quality documents. Elaborate study on poor-quality documents are not undertaken by the scientists in the development of Indian script OCR. Experiments should be made to observe the effect of poor quality paper as well as noise of various types, and take corrective measures.

REFERENCES

[1] S. Marinai "Introduction to document analysis and recognition", Studies in Computational Intelligence (SCI), Vol. 90, pp. 1–20, 2008.

[2] Y.Y. Tang, C.Y. Suen, C.D. Yan, and M.Cheriet, "Document analysis and understanding: a brief survey" First Int. Conf. on Document Analysis and Recognition, Saint-Malo, France, pp. 17-31, October 1991.

[3] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: A comprehensive survey", IEEE Trans on PAMI, Vol.22, pp.62-84, 2000.

[4] Swapan Kr. Parui and Bikash Shaw, "Offline handwritten Devanagari word recognition: An HMM based approach", LNCS 4815, Springer-Verlag, (PReMI-2007), 2007, pp. 528–535.

[5] I. K. Sethi and B. Chatterjee, "Machine recognition of constrained hand printed Devanagari", Pattern Recognition, Vol. 9, pp. 69-75, 1977.

[6] Bikash Shaw, Swapan Kumar Parui and Malayappan Shridhar, "A segmentation based approach to offline handwritten Devanagari word recognition," PReMI, IEEE, pp. 528-35.

[7] B.B. Chaudhuri and A. Majumdar, "Curvelet–based multi SVM recognizer for offline handwritten Bangla: A major Indian script," Int. Conf. of Document And Recognition, 2007, pp. 491-495.

[8] P.S. Deshpande, L. Malik and S. Arora, "Characterizing handwritten Devanagari characters using evolved regular expressions", in Proceeding of TENCON, 2006, pp. 1-4.

[9] A. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[10] N. Otsu, "A threshold selection method from grey level histogram", IEEE Trans on SMC, Vol.9, pp.62-66, 1979.

[11] E. Candes, L. Demanet, D. Donoho and L. Ying, "Fast discrete curvelet transforms,"http://www.curvelet.org/papers/FDCT.pdf

[12] Abraham, B. and Merola, "Dimensionality reduction approach to multivariate prediction," In Esposito Vinzi V. et al. (Eds.): PLS and related methods, CISIA, pp. 3-17.

[13] De Jong, S. and Kiers, H.A.L. "Principal covariates regression", Part-I. Theory, Chemometrics and Intelligent Laboratory Systems, Vol. 14, pp. 155-164.

[14] I.T. Jolliffe "Principal component analysis" Springer series in statistics, 2nd ed., NY, 2002, 487 p. 28 illus. ISBN 978-0-387-95442.

[15] R. O. Duda, P.E. Hart, and D. G. Stork "Pattern classification" John Willy publication.

[16] V. Vapnik, "The nature of statistical learning theory", Springer Verlang, 1995.

[17] Y. Yang, "Expert network: Effective and efficient learning from human decisions in text categorization and retrieval," In Proceeding of 17th. Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94), 1994, pp. 13-22