



Handwritten Devanagari Character Recognition Using Gradient Features

Ashutosh Aggarwal, Rajneesh Rani, RenuDhir
Department of Computer Science and Engineering
Dr B.R. Ambedkar National Institute of Technology
Jalandhar- 144011, Punjab (India)
er.ashutoshaggarwal@gmail.com

Abstract—We describe novel methods of feature extraction for recognition of single isolated Devanagari character images. Our approach is flexible in that the same algorithms can be used, without modification, for feature extraction in a variety of OCR problems. These include handwritten, machine-print, grayscale, and binary and low-resolution character recognition. We use the gradient representation as the basis for extraction of features. These algorithms require a few simple arithmetic operations per image pixel which makes them suitable for real-time applications. Our dataset consists of 200 samples of each of 36 basic Devanagari characters, which are collected from 20 different writers each contributing to write 10 samples of each of 36 characters. Thus we have used total 7200 character samples. All sample images of Devanagari characters used are normalized to 90*90 pixel sizes. A description of the algorithm and experiment with our data set is presented in this paper. Experimental results using Support Vector Machines (SVM) are presented. Our results demonstrate high performance of these features with cross validation accuracy of 94%.

Keywords—Isolated Handwritten Devanagari Character Recognition, Gradient Features, Gradient Feature Extraction, SVM Classifier.

I. INTRODUCTION

Machine simulation of human reading has become a topic of serious research since the introduction of digital computers. The main reason for such an effort was not only the challenges in simulating human reading but also the possibility of efficient applications in which the data present on paper documents has to be transferred into machine-readable format. Automatic recognition of printed and handwritten information present on documents like cheques, envelopes, forms, and other manuscripts has a variety of practical and commercial applications in banks, post offices, libraries, and publishing houses. Optical Character Recognition (OCR) is a field of research in pattern recognition, artificial intelligence and machine vision. OCR is a mechanism to convert machine printed or handwritten document file into editable text format. This field is broadly divided into two parts, Online and offline character recognition. Off-line Character recognition further divided into two parts, machine printed and handwritten character recognition. In handwritten Character Recognition, there are lots of problems as compared to machine printed document because different peoples have different writing styles, the size of pen-tip and some people have skewness in their writing. All this challenges make the researches to solve the problems.

India is a multi-lingual multi-script country and there are twenty two languages. Eleven scripts are used to write these languages and Devanagari Script is an oldest one that is used to write many languages such as Hindi, Nepali, Marathi,

Sindhi and Sanskrit where Hindi is the third most popular language in the world and it is the national language of the India [1]. 300 million people use the Devanagari Script for documentation in central and northern parts of India [2]. A detailed survey report on various work conducted on recognition of Indian Scripts is represented in [14].

The script has a complex composition of its constituent symbols. Devanagari script (Hindi) has 13 vowels and 36 consonants shown in the Fig.1. They are called basic characters. Vowels can be written as independent letters, or by using a variety of diacritical marks which are written above, below, before or after the consonant they belong to. When vowels are written in this way they are known as modifiers and the characters so formed are called conjuncts. Sometimes two or more consonants can combine and take new shapes. These new shape clusters are known as compound characters. All the characters have a horizontal line at the upper part, known as Shirorekha or headline. No English character has such characteristic and so it can be taken as a distinguishable feature to extract English from these scripts. In continuous handwriting, from left to right direction, the Shirorekha of one character joins with the Shirorekha of the previous or next of the same word. In this fashion, multiple characters and modified shapes in a word appear as a single connected component joined through the common Shirorekha. Also in Devanagari there are vowels, consonants, vowel modifiers and component characters, numerals. Moreover, there are many similar shaped characters. All these variations make the handwritten character recognition, a challenging problem.

In our proposed approach, initially the Gradient Vector is calculated at all image pixels and sample image is divided into 9x9 sub-blocks. Then in each sub-block Strength of Gradient is accumulated in each of 8 standard directions in which Gradient Direction is decomposed. Finally image is

down sampled to 5x5 blocks from 9x9 blocks using a Gaussian Filter giving a feature vector of dimensionality 200 (5x5x8). Accuracy of 94% is obtained using Support vector Machines (SVM) as classifier.

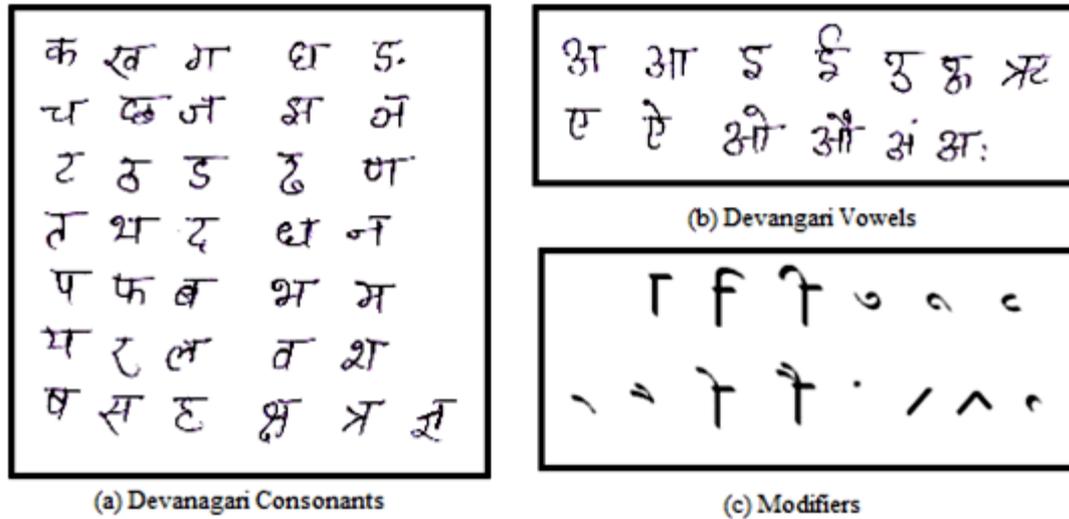


Fig. 1: Devanagari Isolated Handwritten Characters, Modifiers

An overview of the paper is as follows: In Section II, an insight is provided into the earlier work done in recognition of Handwritten Devanagari Character. Section III, covers our proposed approach for Devanagari Character Recognition right from pre-processing of images to use of Gradient as our Feature Extraction Technique and finally a brief introduction of SVM classifier. In Section IV, Experimental results and analysis are provided. Finally, the conclusion & future work have been offered in Section V.

II. RELATED WORK

In literature survey, we found that many researchers had done work towards the off-line handwritten Devanagari character recognition. The first research work report on handwritten Devanagari characters was published in 1977. After that researchers started working on the recognition of handwritten Devanagari characters & tried to solve the problem associated with them. Features used by Sharma et al. [3] for handwritten Devanagari characters are obtained from the directional chain code information of the contour points of the characters. The bounding box of a character is segmented into blocks and a CH (Chain code histogram) is computed in each of the blocks. Based on the CH, they have used 64-D features for recognition. Sharma et al. proposed a quadratic classifier-based scheme for the recognition of handwritten characters and obtained 80.36 % accuracy with the 11,270 dataset size. The work reported in [4] discusses the use of regular expressions (RE) in handwritten Devanagari character recognition, where a hand-written character is converted into an encoded string based on chain-code features. Then, RE of stored templates is matched with it. Rejected samples are then sent to a MED (minimum edit distance) classifier for recognition. On the

5000 samples, this work has been done and 82 % accuracy has been reported. The distortions in handwritten Devanagari characters are removed in [5] using a thickening process followed by thinning and pruning operations. The features are represented using normalized vector distances for each character. The Shirorekha and spine in a handwritten character are detected using a differential-distance-based technique. 50000 samples is used and 89.12 % accuracy has obtained. The recognition of handwritten characters in [6] is based on the modified exponential membership function fitted to the fuzzy sets derived from the features of the characters. A Reuse Policy that provides guidance from the past policies is also utilized in the paper to improve the speed of the learning process and obtained 90.65 % accuracy. For the recognition of handwritten Devanagari non compound characters, shadow features, and CH features are computed in [7]. Two MLPs and a minimum edit distance (MED) method are used for classification of handwritten Devanagari non compound characters in [7]. In the first stage of classification, characters with distinct shapes are classified using two MLPs. Shadow features are used for one MLP and CH features are used for the other MLP for classification. In the second stage of classification, confused characters having similar shapes are classified using a MED method. This method makes use of corners detected in a character image using modified Harris corner detection technique. Kumar [8] compared performances of five feature-extraction methods on handwritten characters. The various features covered are Kirsch directional edges, distance transform, chain code, gradient and directional distance distribution. From the experimentations, it is found that Kirsch directional edges are least performing and gradient is best

performing with SVM classifiers. With multilayer perceptron (MLP), the performance of gradient and directional distance distribution is almost same. The chain-code-based feature is better as compared to Kirsch directional edges and distance transform. A new feature is also proposed in the paper, where the gradient direction is quantized into four-directional levels and each gradient map is divided into 4×4 regions. This is combined with total distances in four directions and neighborhood pixels weight. The features used by Pal et al. [9] for handwritten characters are mainly based on directional information obtained from study of Devanagari handwritten character recognition using 12 different classifiers and four sets of features is presented. A detailed survey report on various techniques used for recognition of both handwritten and machine printed Devanagari Characters have been presented in [13].

III. PROPOSED SYSTEM

A. Dataset preparation & preprocessing

There is no standard dataset available for Devanagari handwritten characters. So dataset is prepared from handwriting of 20 different people who belongs to different age groups. In this work we only used Devanagari consonants. 10 samples of each Devanagari consonant have been written by each people means each people have written 360 (10*36) Devanagari characters in A4 size sheet. After that this sheet is scanned and saved as jpeg image (grayscale image).

Following steps have been performed in order to preprocess the image before feature extraction:

- Intensity values of an image were adjusted.
- Images were converted into binary images by choosing threshold value 0.8.
- All connected components (objects) that have fewer than 30 pixels were removed from the binary images.
- Median filtering, which is a nonlinear operation often used in image processing to reduce "salt and pepper" noise was applied on all images.
- Each Image was segmented horizontally by finding the black pixel in each row.
- After horizontal segmentation, each line was segmented vertically and we obtained our required character image.
- Finally all images of Devanagari characters were normalized to size 90×90 .

B. Feature extraction techniques

Feature extraction is an integral part of any recognition system. The aim of feature extraction is to describe the pattern by means of minimum number of features that are effective in discriminating pattern classes.

i. Gradient Feature Extraction

The gradient measures the magnitude and direction of the greatest change in intensity in a small neighborhood of each pixel. (In what follows, "gradient" refers to both the gradient magnitude and direction). Gradients are computed by means of the Sobel operator. The Sobel templates used to compute the horizontal (X) & vertical (Y) components of the gradient are shown in Fig.2.

the arc tangent of the gradient and Gaussian filter. A modified quadratic classifier is applied on the features of handwritten characters for recognition. Elastic matching (EM) technique based on an Eigen deformation (ED) for recognition of handwritten Devanagari characters is proposed in [10]. In [11], two classifiers are combined to get higher accuracy of character recognition with the gradient features. Combined use of SVM and MQDF is applied for the same. Many approaches have been proposed toward handwritten Devanagari numeral, character, and word recognition in the past decade [12]. A comparative

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Horizontal Component Vertical Component

Fig. 2: Sobel masks for gradient

Given an input image of size $D_1 \times D_2$, each pixel neighbourhood is convolved with these templates to determine these X and Y components, S_x and S_y , respectively. Eq. (1) and (2) represents their mathematical representation :

$$S_x(i, j) = I(i-1, j+1) + 2 * I(i, j+1) + I(i+1, j+1) - I(i-1, j-1) - 2 * I(i, j-1) - I(i+1, j-1). \quad (1)$$

$$S_y(i, j) = I(i-1, j-1) + 2 * I(i-1, j) + I(i-1, j+1) - I(i+1, j-1) - 2 * I(i+1, j) - I(i+1, j+1) \quad (2)$$

Here, (i,j) range over the image rows (D_1) and columns (D_2), respectively. The gradient strength and direction can be computed from the gradient vector $[S_x, S_y]^T$ as shown below using Eq. (3) and (4):

The gradient magnitude is then calculated as:

$$r(i, j) = \sqrt{S_x^2(i, j) + S_y^2(i, j)} \quad (3)$$

Gradient direction is calculated as:

$$\theta(i, j) = \tan^{-1} \frac{S_y(i, j)}{S_x(i, j)} \quad (4)$$

After obtaining gradient vector of each pixel, the gradient image is decomposed into four orientation planes or eight direction planes (chaincode directions) as shown in Fig.3.

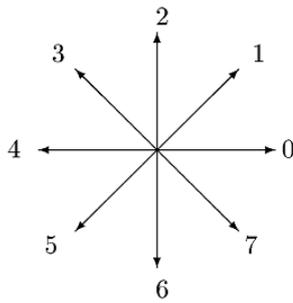


Fig. 3: 8 directions of Chaincodes

After this, gradient vector of each pixel is decomposed into components along these standard direction planes. If a gradient direction lies between two standard directions, it is

ii. Generation of Gradient Feature Vector

A gradient feature vector is composed of the strength of gradient accumulated separately in different directions as described below:

- (1) The direction of gradient detected as above is decomposed along 8 chaincode directions.
- (2) The character image is divided into 81(9 horizontal × 9 vertical) blocks. The strength of the gradient is accumulated separately in each of 8 directions, in each block, to produce 81 local spectra of direction.
- (3) The spatial resolution is reduced from 9×9 to 5×5 by down sampling every two horizontal and every two vertical blocks with 5 × 5 Gaussian Filter to produce a feature vector of size 200 (5 horizontal, 5 vertical, 8 directional resolution).
- (5) The variable transformation ($y = x^{0.4}$) is applied to make the distribution of the features Gaussian-like.

The 5 × 5 Gaussian Filter used is the high cut filter to reduce the aliasing due to the down sampling as done in paper [15].

C. Classifier (SVM)

Support Vector Machine is supervised Machine Learning technique. It is primarily a two class classifier. Width of the margin between the classes is the optimization criterion, i.e. the empty area around the decision boundary defined by the distance to the nearest training pattern. These patterns called support vectors, finally define the classification function. All the experiments are done on LIBSVM 3.0.1[20] which is multiclass SVM and select RBF (Radial Basis Function) kernel. A feature vector set $f_v(x_i) = 1 \dots m$, where m is the total number of character in training set and a class set $cs(y_j) = 1 \dots m$, $cs(y_j) \in \{0, 1, \dots, 9\}$ which defines the class of the training set, fed to Multi Class SVM.

LIBSVM implements the “one against one” approach (Knerr et al., 1990) [17] for multi-class classification. Some early works of applying this strategy to SVM include, for example, Kressel (1998) [16]. If k is the number of classes, then $k(k-1)/2$ classifiers are constructed and each one trains data from two classes. For training data from the i^{th} and j^{th} classes, we solve the following two class classification problem:

$$\min_{w^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + c \sum_t (\xi^{ij})_t,$$

decomposed into two components in the two standard directions, as shown in Fig.4.

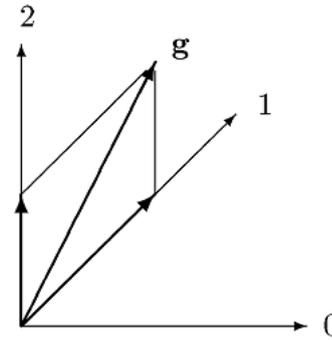


Fig. 4: Decomposition of gradient direction

$$\begin{aligned} &\text{subject to } (w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \\ &\quad \text{if } x_t \text{ in the } i^{th} \text{ class} \\ &(w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \\ &\quad \text{if } x_t \text{ in the } j^{th} \text{ class,} \\ &\quad \xi_t^{ij} \geq 0. \end{aligned}$$

According to how all the samples can be classified in different classes with appropriate margin, different types of kernel in SVM classifier are used. Commonly used kernels are:

Linear kernel, Polynomial kernel, Gaussian Radial Basis Function (RBF) and Sigmoid (hyperbolic tangent).

The effectiveness of SVM depends on kernel used, kernel parameters and soft margin or penalty parameter C . The common choice is RBF kernel, which has a single parameter γ (g or γ). We also have selected RBF kernel for our experiment.

Radial Basis Function (RBF) kernel, denoted as

$$K(x_i, x_j) \equiv \exp(-\gamma \|x_i - x_j\|^C)$$

Best combination of C and γ for optimal result is obtained by grid search by exponentially growing sequence of C and γ and each combination is cross validated and finally parameters in combination giving highest cross validation accuracy are selected as optimal parameters.

In k -fold cross validation we first divide the training set into k equal subsets. Then one subset is used to test by classifier trained by other remaining $k-1$ subsets. By cross validation each sample of train data is predicted and it gives the percentage of correctly recognized dataset.

IV. EXPERIMENTS AND RESULTS

In order to classify the handwritten character and evaluate the performance of the technique, we have carried out the experiment by setting parameters γ , and cost parameter. All experiments was performed on a Intel® core 2 duo CPU T6400 @ 2GHz with 3 GB RAM under 64 bit windows 7 Ultimate operating system.

5 fold cross validation is applied for recognition accuracy. We experiment with different γ values of the γ as shown in Table 1 and obtained 94 % recognition rate at the value of $\gamma = 0.4$ and cost (c) = 500.

While observing the results at other values of parameter C, it is analyzed that decreasing the value of C irrespective of any change in γ slightly decreases the recognition rate, but on increasing the value of C and after a certain increment normally after 64 i.e. at higher values of C the recognition rate becomes stable. In contrast, the recognition rate always changes with the change in γ .

Table 1: shows the recognition accuracy

V. CONCLUSION & FUTURE WORK

In the literature, many techniques for recognition of Devanagari Handwritten Characters have been suggested. In this paper an effort is made towards recognition of Devanagari Characters and obtained recognition accuracy of 94%. Due to its logical simplicity, ease of use and high recognition rate, Gradient Features should be used for recognition purposes in other Indian Scripts like Gurmukhi, Malayalam etc. where not much research is conducted for their recognition. More research work should be conducted in using Gradient Features in combination with other feature

REFERENCES

- [1] U.Pal and B B Choudhuri, "Indian script character recognition: A survey" Pattern Recognition, Vol 37, pp 1887-1899, 2004.
- [2] R.M.K.Sinha, "A journey from Indian scripts processing to Indian language processing", IEEE Ann. Hist. Computer, vol 31, no 1, pp 831, 2009.
- [3] N. Sharma, U. Pal, F. Kimura, and S. Pal, "Recognition of offline hand-written Devanagari characters using quadratic classifier," in Proc. Indian Conference on Computer, Vision and Graphics, Image Processing 2006, pp. 805-816.
- [4] P.S.Deshpande, L.Malik, S.Arora, "Fine classification & recognition of hand written Devanagari characters with regular expressions & minimum edit distance method," J. Comput., vol. 3, no. 5, pp. 11-17, 2008.
- [5] S.Arora, D.Bhattacharjee, M.Nasipuri, and L.Malik, "A two stage classification approach for handwritten Devanagari characters," in Proceedings of International Conference on Computer Intelligence & Multimedia Applications, 2007, pp. 399-403.
- [6] M.Hanmandlu, O.V.R.Murthy, V. K.Madasu, "Fuzzy Model based recognition of handwritten Hindi characters," in Proceedings of International Conference Digital Image Comput. Tech. Appl., 2007, pp. 454-461.
- [7] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, and M. Kundu, "Recognition of non-compound handwritten Devanagari characters using a combination of MLP and minimum edit distance," International Journal of Computer Science and Security, vol. 4, no. 1, pp. 1-14, 2010.
- [8] S. Kumar, "Performance comparison of features on Devanagari hand-printed dataset," International Journal Recent Trends, vol. 1, no. 2, pp. 33-37, 2009.
- [9] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Off-line handwritten character recognition of Devanagari script," in Proc. 9th Conf. Document Analysis and Recognition, 2007, pp. 496-500.
- [10] V. Mane, L. Raha, "Handwritten character recognition using elastic matching and PCA," in Proceedings of International Conference on Computer Commun. Control, 2009, pp. 410-415.
- [11] U. Pal, S. Chanda, T. Wakabayashi, and F. Kimura, "Accuracy improvement of Devanagari character recognition combining SVM and MQDF," in Proc. 11th Int. Conf. Frontiers Handwritten Recognition, 2008, pp. 367-372.
- [12] U. Pal, T. Wakabayashi, and F. Kimura, "Comparative study of Devanagari handwritten character recognition using different features and classifiers," in Proc. 10th Conf. Document Analysis Recognition, 2009, pp. 1111-1115.
- [13] R. Jayadevan ET. Al. "Offline Recognition of Devanagari Script: A Survey", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 41, No. 6, November 2011.
- [14] U. Pal, N. Sharma, R. Jayadevan "Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques," ACM Transactions on Asian Language Information Processing, Vol. 11, No. 1, Article 1, Publication date: March 2012.
- [15] Akinoria Kawamura et.al "Online Recognition of Freely handwritten Japanese Characters Using Directional feature Densities" IEEE XPLore Digital Library 1992.
- [16] U. H.-G. Kressel. "Pairwise classification and support vector machines. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods { Support Vector Learning, pages 255-268, Cambridge, MA, 1998. MIT Press }.
- [17] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: a stepwise procedure for

extraction techniques & different-2 classifiers in order to improve recognition accuracy.

S. No	5-folds, Dataset size = 7200, Cost(c)=500,	
	Gamma (γ)	Recognition Accuracy
1	0.1	89.72 %
2	0.2	90.83 %
3	0.3	92.84 %
4	0.4	94 %
5	0.5	93.13 %
6	0.6	90.89 %
7	0.7	88.86 %
9	0.9	87.82 %
10	1	86.81 %

building and training a neural network. In J. Fogelman, editor, Neurocomputing: Algorithms, Architectures and Applications. Springer-Verlag, 1990.

[18] <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

[19] <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm>

[20] <http://www.isical.ac.in/~ujjwal/download/datab ase.html>.