Robust Printed Devanagari Document Recognition using Hybrid Approach of Shirorekha Chopping, Fuzzy Directional Features and Support Vector Machine

Nitin Mishra
Tata Consultancy Services Limited
Hyderabad, Andhra Pradesh, India

Ankur Agrawal
Tata Consultancy Services Limited
Hyderabad, Andhra Pradesh, India

ABSTRACT

This paper presents a novel methodology for recognizing machine printed Devanagari script document. Shirorekha Chopping based preprocessing is chosen to enable the segmentation of printed text into various characters. Fuzzy Directional Features have shown improvement over commonly used Directional features. A set of 8 directional Fuzzy Directional Features (FDF) for each character is extracted and classified to the appropriate character class. Radial Basis function (RBF) kernel based Support Vector Machines (SVM) model is used for training the various multi font characters and testing the Devanagari document to be recognized. Experiments are conducted for the multi font Devanagari document recognition. The recognition rate of the proposed OCR system with the image document of Devnagari Script has been found to be 97.9% for two fonts Mangal and Krutidev.

General Terms

Pattern Recognition, Image Processing and Computer Vision

Keywords

Devanagari, OCR, Shirorekha Chopping, Fuzzy Directional Features, Support Vector Machine.

1. INTRODUCTION

Information extraction from printed documents plays a key role in many areas: office automation, knowledge management, intelligence and so on. Manual processing is often not feasible or fully satisfactory, typically because of the high volume of documents to be processed and the high cost per-unit introduced by human operators. Augmenting the automation degree of information extraction from printed documents may thus have a substantial impact on many settings of highly practical interest. English Character Recognition (CR) has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. But same is not the case for Indian languages which are complicated in terms of structure and computations. Rapidly growing computational power may enable the implementation of Indic CR methodologies. Since Hindi language has enormous number of character combinations [1, 2]; it is not a good technique to train all the possible combinations of Hindi characters. It is highly desirable to judiciously choose a Database having all basic characters, half characters, and the minimal set of conjunct character combinations that may occur in some word and leave out all unfavorable combinations based on stochastic Finite Automaton approach. The proposed Hindi Language Database consists of basic vowels, consonants, special symbols, punctuation marks, English numerals, Devnagari numerals and minimal set of favorable vowel consonant combinations, bi-consonant combinations and biconsonant-vowel combinations [3]. Fuzzy Directional Features (FDFs) have shown great improvement over common directional features in OCR. Recently RBF kernel based Support Vector machines (SVMs) have become popular in data classification techniques [4, 5]. Proposed methodology makes use of RBF kernel based SVM with 5-fold cross validation. LIBSVM tool is used to train and test the Fuzzy Directional Features of 486 character classes consisting of 15 vowels, 36 consonants, 13 special symbols, 18 punctuation marks and other symbols, 10 English numerals, 10 Devnagari numerals, a minimal set of 116 vowel-consonant combinations, 183 bi-consonant combinations and 79 bi-consonant-vowel combinations.

2. METHODOLOGY

The Proposed Methodology as shown in Figure 1 applies Shirorekha Chopping on the binarized and de-skewed printed Devanagari document image and segments the characters from it. The Fuzzy Directional Features are extracted from the

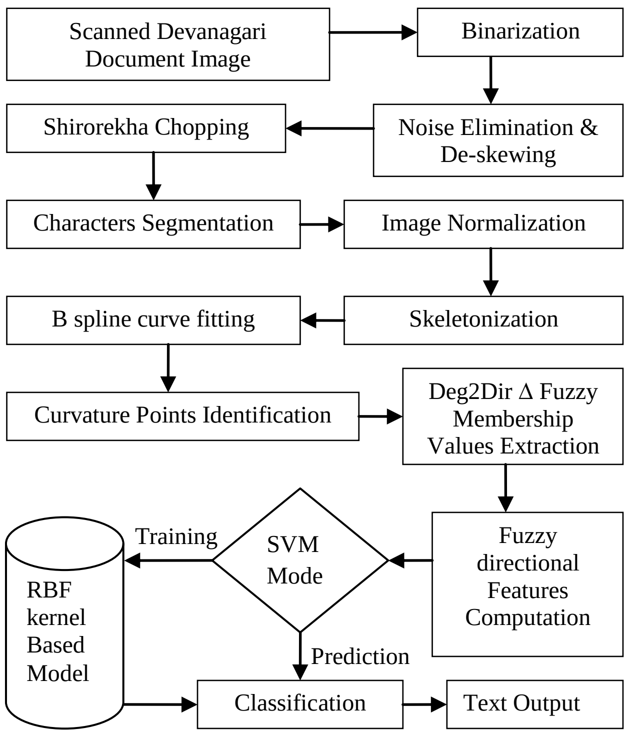

Fig 1: Block Diagram of Proposed Method

The segmentation issues related to Shirorekha based scripts are presented in [6, 7]. The Segmented Characters are trained and predicted using Radial Basis Function kernel based SVM model with 5–fold cross validation based automated parameters decision done by LIBSVM tool. The results show that the Proposed Approach for Devanagari OCR is quite efficient.

Table 1: General Vowels

अ a	आ ा aa/A	इ ि e/i	ई ी ee/ii	उ ु u	ऊ ू oo/uu
ए े e	ऐ ै ai	ओ ो o	औ ौ ou	अं ं aM	अः ः aH

Table 2: Other Vowels

ॠ r^^	ॡ l^^	ॐ AUM

Table 3: Consonants

क ka	ख kha	ग ga	घ gha	ङ nga
च cha	छ chha	ज ja	झ jha	ञ nja
ट Ta	ठ Tha	ड Da	ढ Dha	ण Na
त ta	थ tha	द da	ध dha	न na
प pa	फ Pha/fa	ब ba	भ bha	म ma
य ya	र ra	ल la	व va/wa	श Sha
ष shh	स sa	ह ha	क्ष ksh	त्र tra
ज्ञ jnja				

Table 4: Special Symbols

Anusvara ं	Visarga ः	Chandra Bindu ँ	Chandra ॅ
Nukta ़	Virama ्	Udatta ॑	Anudatta ॒
Purna virama ।	Deergha virama ॥	Avagraha ऽ	Grave Accent ॓
Accute Accent ॔			

Table 5: Punctuation Marks and Other Symbols

“	?	;	%	*	/	(	)	\
=	{	}	[	]	,	-	:	!

Table 6: Numerals

०	१	२	३	४	५	६	७	८	९
0	1	2	3	4	5	6	7	8	9

3. SHIROREKHA CHOPPING

The global horizontal projection method computes sum of all black pixels on every row and constructs corresponding histogram. Based on the peak/valley points of the histogram, individual lines, words and characters are segmented.

3.1 Shirorekha Chopping Algorithm

The steps of Shirorekha Chopping Algorithm are as follows:

Input: Binarized Image of Devanagari Document

Output: Normalized Segmented Character Images

3.1.1 Line Segmentation

In line segmentation the aim is to draw one upper horizontal line and one lower horizontal line for each line of text image. The steps for line segmentation are as follow:

1) Construct the Horizontal Histogram for the image.
2) Using the Histogram, find the points from which the line starts and ends.
3) For a line of text, upper line is drawn at a point where we start finding black pixels and lower line is drawn where we start finding absence of black pixels. And the process continues for next line and so on.

3.1.2 *Word Segmentation*

The steps for word segmentation are as follow:

1) Construct the vertical histogram for each segmented line.
2) Using the vertical Histogram, find the points from which the word starts and ends.
3) Vertical lines are drawn at starting and ending points for each word.

3.1.3 *Character Segmentation*

The steps for character segmentation are as follow:

1) Draw the horizontal histogram for each segmented line.
2) From the horizontal histogram, find the row which consists of maximum value.
3) The row which consists of maximum value of black pixel for each line is actually the row which consists of Header line also known as Shirorekha.
4) Draw the vertical histogram for each segmented word.
5) Using the histogram, find the points from which the character starts and ends.
6) Draw line according these coordinate.
7) Chop the Shirorekha at points at which distance between bottom of valley and x-axis of Vertical histogram becomes less than or equal to width of Header line.

3.1.4 *Bounding Box Generation*

1) Maintain the data structure to feed the line, word and character boundaries such that the character boundary could be sufficiently extracted from the image so that the bounding box is generated around each segmented character.

3.1.5 *Segmented Character Image Generation*

1) Each bounding box is cropped and segmented character images are normalized to same resolution.
2) **Return** final character images.

The concept of Shirorekha Chopping is illustrated in Fig 2.

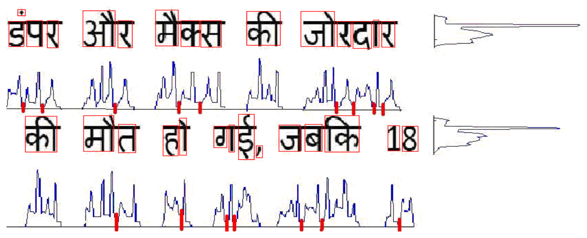

Fig 2: Shirorekha Chopping

4. IDENTIFYING CURVATURE POINTS

Curvature points specify the points at which relative gradient changes thus have a vital role in Fuzzy Directional Features.

4.1 Curvature Points Identification Algorithm

The steps of Curvature Points Identification Algorithm are as follows:

```
Input: Skeleton of Character Image
Output: Series of Curvature points $C= \{c_1, c_2, \ldots\ldots\ldots , c_k\}$
1)  Let the Skeleton be represented by a series of 2D points
    as $\{(x_1, y_1), (x_2, y_2), \ldots\ldots , (x_n, y_n)\}$
2)  For i=1 to n-1 do
    a)  if $(x_i - x_{i+1}) > 0$
        i)  $x_i' = +1$
    b)  else if $(x_i - x_{i+1}) < 0$
        i)  $x_i' = -1$
    c)  else if $(x_i - x_{i+1}) = 0$
        i)  $x_i' = 0$
    End
3)  For i=1 to n-1 do
    a)  if $(y_i - y_{i+1}) > 0$
        i)  $y_i' = +1$
    b)  else if $(y_i - y_{i+1}) < 0$
        i)  $y_i' = -1$
    c)  else if $(y_i - y_{i+1}) = 0$
        i)  $y_i' = 0$
    End
4)  For i=1 to n-2 do
    a)  if $(x_i' - x_{i+1}') \neq 0$ OR $(y_i' - y_{i+1}') \neq 0$
        i)  Point $(x_i, y_i)$ is a curvature point, $c_i$
    End
5)  Return the series of curvature points, $C= \{c_1, c_2, \ldots , c_k\}$
    Where Point $(x_i, y_i)$ is a curvature point, $c_i$ moreover $c_i$
    and $c_{i+1}$ are adjacent curvature points.
```

5. FUZZY DIRECTIONAL FEATURES

FDF are simple and effective feature set. Let θ_i be the angle between two adjacent curvature points c_i and c_{i+1}. Divide 2π into 8 overlapping directions so that every θ_i has two directions d_i^1 and d_i^2 adjacent to each other. i.e., $d_i^1 = \mathbf{1}$ and $d_i^2 = \mathbf{2}$, with its associated membership values $\in [0, 1]$.

i.e., $m_i^1 + m_i^2 = 1.0$.

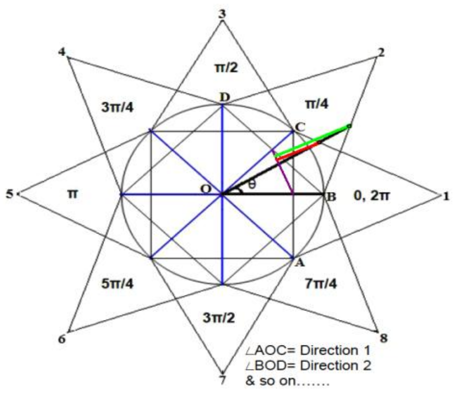

Fig 3: Fuzzy Directional Features

5.1 Triangular Fuzzy Membership Values Extraction Algorithm

The steps of the Algorithm are as follows:

Input: Series of Curvature Points, C= {c_1, c_2, ……… , c_k}

Output: Matrix, m of size (k-1) x 8.

```
1)  Initialize Matrix, m of size (k-1) x 8 with initial value 0.0
2)  Let $m_i^j$ be entry at ith row and jth column in the matrix, m.
3)  For i=1 to k-1 do
    a)  $θ_i = \tan^{-1}((y_i - y_{i+1}) / (x_i - x_{i+1}))$
    b)  if $θ_i$ Є [ 7π/4, π/4)              // CASE: 1
        i)   $m_i^1$= 1.0-(abs(0 - $θ_i$ )/ (π/4))
        ii)  if $m_i^1 \neq 1.0$ AND $θ_i$ <2π
                 $m_i^8$= 1.0-(abs(7π/4 - $θ_i$ )/ (π/4))
             else if $m_i^1 \neq 1.0$ AND $θ_i$ >0
                 $m_i^2$= 1.0-(abs(π/4 - $θ_i$ )/ (π/4))
    c)  else if $θ_i$ Є [ 0, π/2 )           // CASE: 2
        i)   $m_i^2$= 1.0-(abs(π/4 - $θ_i$ )/ (π/4))
        ii)  if $m_i^2 \neq 1.0$ AND $θ_i$ <π/4
                 $m_i^1$= 1.0-(abs(0 - $θ_i$ )/ (π/4))
             else if $m_i^2 \neq 1.0$ AND $θ_i$ >π/4
                 $m_i^3$= 1.0-(abs(π/2 - $θ_i$ )/ (π/4))
    d)  else if $θ_i$ Є [ π/4, 3π/4 )        // CASE: 3
        i)   $m_i^3$= 1.0-(abs(π/2 - $θ_i$ )/ (π/4))
        ii)  if $m_i^3 \neq 1.0$ AND $θ_i$ <π/2
                 $m_i^2$= 1.0-(abs(π/4 - $θ_i$ )/ (π/4))
             else if $m_i^3 \neq 1.0$ AND $θ_i$ >π/2
                 $m_i^4$= 1.0-(abs(3π/4 - $θ_i$ )/ (π/4))
    e)  else if $θ_i$ Є [ π/2, π )           // CASE: 4
        i)   $m_i^4$= 1.0-(abs(3π/4 - $θ_i$ )/ (π/4))
        ii)  if $m_i^4 \neq 1.0$ AND $θ_i$ <3π/4
                 $m_i^3$= 1.0-(abs(π/2 - $θ_i$ )/ (π/4))
             else if $m_i^4 \neq 1.0$ AND $θ_i$ >3π/4
                 $m_i^5$= 1.0-(abs(π - $θ_i$ )/ (π/4))
    f)  else if $θ_i$ Є [ 3π/4, 5π/4 )       // CASE: 5
        i)   $m_i^5$= 1.0-(abs(π - $θ_i$ )/ (π/4))
        ii)  if $m_i^5 \neq 1.0$ AND $θ_i$ <π
                 $m_i^4$= 1.0-(abs(3π/4 - $θ_i$ )/ (π/4))
             else if $m_i^5 \neq 1.0$ AND $θ_i$ >π
                 $m_i^6$= 1.0-(abs(5π/4 - $θ_i$ )/ (π/4))
    g)  else if $θ_i$ Є [ π, 3π/2 )          // CASE: 6
        i)   $m_i^6$= 1.0-(abs(5π/4 - $θ_i$ )/ (π/4))
        ii)  if $m_i^6 \neq 1.0$ AND $θ_i$ <5π/4
                 $m_i^5$= 1.0-(abs(π - $θ_i$ )/ (π/4))
             else if $m_i^6 \neq 1.0$ AND $θ_i$ >5π/4
                 $m_i^7$= 1.0-(abs(3π/2 - $θ_i$ )/ (π/4))
    h)  else if $θ_i$ Є [ 5π/4, 7π/4 )       // CASE: 7
        i)   $m_i^7$= 1.0-(abs(3π/2 - $θ_i$ )/ (π/4))
        ii)  if $m_i^7 \neq 1.0$ AND $θ_i$ <3π/2
                 $m_i^6$= 1.0-(abs(5π/4 - $θ_i$ )/ (π/4))
             else if $m_i^7 \neq 1.0$ AND $θ_i$ >3π/2
                 $m_i^8$= 1.0-(abs(7π/4 - $θ_i$ )/ (π/4))
    i)  else if $θ_i$ Є [ 3π/2, 2π )         // CASE: 8
        i)   $m_i^8$= 1.0-(abs(7π/4 - $θ_i$ )/ (π/4))
        ii)  if $m_i^8 \neq 1.0$ AND $θ_i$ <7π/4
                 $m_i^7$= 1.0-(abs(3π/2 - $θ_i$ )/ (π/4))
             else if $m_i^8 \neq 1.0$ AND $θ_i$ >7π/4
                 $m_i^1$= 1.0-(abs(2π - $θ_i$ )/ (π/4))
    End
4)  Return Matrix m having all Fuzzy Membership values.
```

Fuzzy Aspect comes into the picture due to the membership function which associates the angle between two adjacent curvature points into two directions with different membership values. It is Also noted that if the angle $Θ_i$ has 2 directions d^j and d^{j+1} then $m_i^{\mathbf{j}} + m_i^{\mathbf{j+1}}$=1.0, where j Є {1, 2,..8}.

5.2 Computation of Fuzzy Directional Features (FDF)

The fuzzy membership values assigned to each direction are represented as m_1^1, $m_1^2 \cdot \cdot \cdot$, m_{k-1}^8 and these values are corresponding to f_1, . . . , f_8 feature vector values and k curvature points. Here $θ_1, \cdot\cdot\cdot θ_{k-1}$ are the angles between two consecutive curvature points (where k is the total number of curvature points) in a segmented character image. The FDF is calculated by averaging Matrix, m across the columns, so as to form a vector of dimension eight. The mean is calculated as follows; for each direction (1 to 8), collect all the membership values and divide by the number of occurrences of the membership values in that direction as shown in **Fig 4**. In all the experiments these mean values were used to construct the 8 directional Fuzzy Directional Features to represent a Segmented Character .i.e., **F = [f_1, f_2, . . . , f_8]** where,

$$f_1 = \frac{(m_2^1 + m_3^1 + m_{k-2}^1)}{3}.$$

6. SUPPORT VECTOR MACHINE

Recently, support vector machines (SVMs) have been a promising tool for the data classification [8]. Its basic idea is to map data into a high dimensional space and find a separating hyperplane with the maximal margin. Given training vectors x_k Є R^n, k = 1, ... ,m in two classes, and a vector of labels y Є R^m such that y_k Є {1,−1}, SVM solves a quadratic optimization problem [9] :

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{k=1}^{m} \xi_k$$

Subject to $y_k\left(w^T\phi\left(x_k\right)+b\right)\geq 1-\xi_k$,

$\xi_k \geq 0$, k = 1, ... ,m

Where training data are mapped to a higher dimensional space by the function ϕ and C is a penalty parameter on the training error. For any testing instance x, the decision function (predictor) is

$$f(x)=\operatorname{sgn}(w^T\phi(x)+b)$$

Practically, we need only $k(x,x')=\phi(x)^T\ \phi(x')$, the kernel function to train the SVM. The **RBF kernel** is used in proposed approach:

$$k(x,x')=\exp(-\gamma\,\|x-x'\|^2)$$

With the RBF kernel there are two parameters to be determined in the SVM model: C and γ. To get good generalization ability validation process is conducted to decide the parameters. The procedure is as follows:

1) Consider a grid space of (C,γ) with $\log_2 C \in \{-5,-3,\dots,15\}$ and $\log_2\gamma \in \{-15,-13,\dots,3\}$

2) For each hyperparameter pair (C,γ) in the search space, conduct 5-fold cross validation on the training set.

3) Choose the parameter (C,γ) that leads to lowest CV balanced error rate.

4) Finally use the best parameter to create a model as the predictor.

LIBSVM tool recommends linearly scaling each feature to the range [-1, +1] or [0, 1] before SVM training or matching phase. As All the Fuzzy Directional Features are in the range [0, 1], there is no need to rescale the features and the experiments are easily conducted using LIBSVM tool [10].

θ↓) d→)	1	2	3	4	5	6	7	8
Θ_1	0	0	m_1^3	m_1^4	0	0	0	0
Θ_2	m_2^1	m_2^2	0	0	0	0	0	0
Θ_3	m_3^1	0	0	0	0	0	0	m_3^8
.								
.								
.								
.								
Θ_i	0	0	0	0	0	0	m_i^7	m_i^8
.								
.								
Θ_{k-2}	m_{k-2}^1	m_{k-2}^2	0	0	0	0	0	0
Θ_{k-1}	0	0	0	0	m_{k-1}^5	m_{k-1}^6	0	0
F	$f_1=$ $\mathbf{(m_2^1+m_3^1+m_{k-2}^1)/3}$	$f_2=$ $\mathbf{(m_2^2+m_{k-2}^2)/2}$	$f_3=$ $\mathbf{m_1^3}$	$f_4=$ $\mathbf{m_1^4}$	$f_5=$ $\mathbf{m_{k-1}^5}$	$f_6=$ $\mathbf{m_{k-1}^6}$	$f_7=$ $\mathbf{m_i^7}$	$f_8=$ $\mathbf{(m_3^8+m_i^8)/2}$

Fig 4: Computation of Fuzzy Directional Features from Matrix m

7. EXPERIMENTAL RESULTS

This section presents the experimental results. After applying the Shirorekha Chopping the characters are easily segmented. The skeletons of the segmented characters are generated and B spline curves are fitted in order to smooth the skeleton and find the curvature points. Once the curvature points in the segmented character image are identified the relative angle between adjacent curvature points are computed and fed into the process of Fuzzy Directional Features extraction.

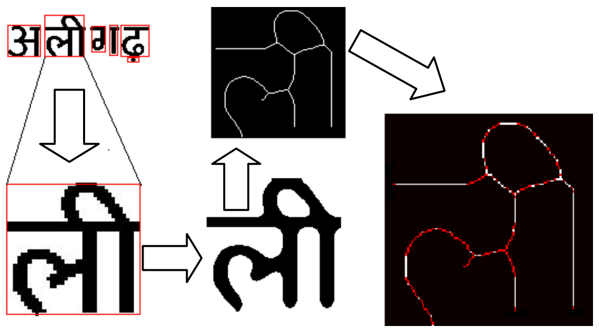

Fig 5: Curvature Points Identification

All FDFs are trained/predicted using RBF kernel based SVM with 5-fold cross validation parameters adjustment. In 5-fold cross validation, training set is divided into 5 subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining 4 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

Table 7: Parameters Selection

	Cost Parameter, C	**Gamma Parameter, g**	**# of Cross Validation, v**
Test 1	5	2.0	0
Test 2	2	1.0	10
Test 3	**0.03125**	**0.0078**	**5**

Table 7 shows the Initial Test for the Best Parameters selections for the training of character classes by LIBSVM. After a number of tests it was observed that Test 3 was most favourable for the training phase. Thus Parameter C, g and v were globally set to 0.03125, 0.0078 and 5 respectively.

Table 8: Characters Classification Test

	Trained classes	**Test samples**	**Correct classifications**
Test A	486	486	459
Test B	486	100	97
Test C	486	1000	989

Table 8 shows the performance of the tests conducted for the Characters Classification. Finally According to the correct classifications and misclassifications, the average recognition accuracy was found to be 97.9%.

Table 9: Recognition Accuracy

	Recognition Accuracy	**Mean Square Error**
Test A	94.4%	0.86
Test B	97.0%	0.425
Test C	98.9%	0.157

8. CONCLUSIONS

There is a significant improvement in the recognition accuracy in the Devanagari OCR after integrating Shirorekha Chopping with Fuzzy Directional Features and Support Vector Machine. Table 9 shows the higher recognition accuracy for SVM based on RBF kernel.

9. REFERENCES

[1] Bansal, V. and Sinha, R.M.K. “A Complete OCR for Printed Hindi Text in Devnagari Script”, Sixth International Conference on Document Analysis and Recognition, IEEE Publication, Seatle USA, 2001, Page(s):800-804.

[2] Jindal, M.K., Sharma, R.K., lehal, G.S. “A Study of Different Kinds of Degradation in Printed Gurmukhi Script”, Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07),2007.

[3] Yadav, D., Sharma, A.K. and Gupta, J.P. Optical character recognition for printed Hindi text in Devanagari using soft-computing technique, *IASTED International Multi-Conference: Artificial Intelligence and Applications, Innsbruck, Austria,* 2007, pp. 102-107

[4] Chaudhuri, B. B. and Pal, U. "An OCR System to Read Two Indian Language Scripts: Bangla and Devnagari (Hindi)", Proc. of 4th ICDAR vol.2, Ulm, Germany, 1997, Page(s): 1011 -1015

[5] Pal, U., Chaudhuri, B. B. "Indian Script Character recognition: A survey", *Pattern Recognition*, vol. 37, pp. 1887-1899, 2004..

[6] Agrawal, P., Hanmandlu, M. and Lall, B., “Coarse Classification of Handwritten Hindi Characters”, International Journal of Advanced Science and Technology,Vol. 10, September, 2009.

[7] Saba, T., Sulong, G. and Rehman, A. “A Survey on Methods and Strategies on Touched Characters Segmentation”, International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 1, No. 2, June 2010.

[8] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. On-line handwriting recognition with support vector machines.a kernel approach. In Proc. of the 8th IWFHR, pages 49.54, 2002.

[9] G. C. Cawley. MATLAB support vector machine toolbox (v0.55¯) University, of East Anglia, School of Information Systems, Norwich, Norfolk, U.K., 2000. available at: [http://theoval.sys.uea.ac.uk/.gcc/svm/toolbox].

[10] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1—27:27, 2011. The Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm