A Study on the Effect of Outliers in Devanagari Character Recognition

O.V. Ramana Murthy
Dept of Electrical Engineering
IIT Delhi
India – 110016

M. Hanmandlu
Dept of Electrical Engineering
IIT Delhi
India – 110016

ABSTRACT
Devanagari is the basic script for many languages of India, including their National language Hindi. Unlike the Latin script used for the English language, it does not have upper case or lowercase. It has only one case of writing. Moreover each alphabet contains more curves than straight lines. Hence handwritten Devanagari character recognition is a challenging task. To capture different handwritten styles of each alphabet, different approaches have been proposed. In this work, we investigate a simple filtering technique on the features. Support Vector Machine (SVM) was used as classifier. It has been applied on two benchmark Devanagari databases and results show an improvement of as much as 5-10%. This improvement is found to be consistent with different sizes of the database. It was studied on pixel density features and GIST features separately. GIST features were found to be more effective and like having the potency of self-containing filtering.

General Terms
Outliers, Support Vector Machine, Character recognition, pixel density features, GIST features.

Keywords
Outliers, Support Vector Machine, Character recognition, pixel density features, GIST features.

1. INTRODUCTION
Devanagari is the basic script for many languages of India, including their National language Hindi. Hindi is also the third most popular language in the world [8]. Several research reports are available for Devanagari off-line handwritten character recognition now. [4] gives good survey on other reports and the techniques they have employed.

Devanagari script has 11 vowels and 33 consonants, as shown in Fig.1. These are called basic characters. All the characters have a horizontal line at the upper part, known as *Shirorekha* or headline. As it can be observed, in any alphabet, the number of curves is more than the number of straight lines. It does not have any upper or lowercase way of writing.

अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ

(a) Vowels

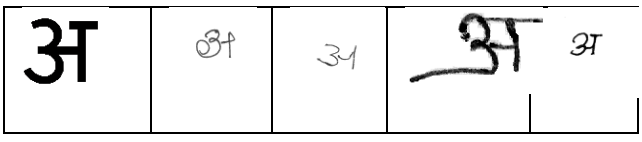
क	ख	ग	घ	ङ
च	छ	ज	झ	ञ
ट	ठ	ड	ढ	ण
त	थ	द	ध	न
प	फ	ब	भ	म
य	र	ल	व	श
ष	स	ह		

(b) Consonants

Fig. 1 Vowels and consonants of Devanagari script

Several studies have proposed different feature extraction techniques and classifier combinations, to show better recognition rates. In this work, we demonstrate that if we can carefully filter the outliers while extracting the features, there can be substantial improvement in the recognition rates. For instance, Fig.2 shows nine different images of handwritten Devanagari character 'अ' that were taken from a benchmark database. Although each image can be recognized as representing 'अ', the variability is very large. Particularly, there is major variation in the curve endings only. We hypothesize that while capturing the features if sufficient care be taken so that the characteristics at these curve endings is proper, better performance can be achieved with the same feature extraction and classifier combination.

The paper is organized as follows. Section-II gives the layout of the overall framework, feature extraction and outliers' detection. Section III gives a very brief review on SVM used as classifier. Section-IV briefly reports the results on handwritten Devanagari Character recognition and its analysis. Finally Section-V draws the conclusions.



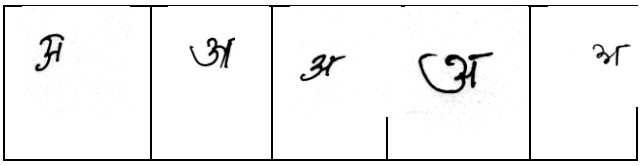


Fig.2 A original machine printed character and 9 different handwritten samples of it

2. OVERALL FRAMEWORK

The overall recognition approach is briefly described here with a schematic diagram shown in Fig. 3. We extract features and then incorporate our proposed filtering technique. There is substantial improvement in the performance when compared to using the features directly. We also compare our results with the other benchmark results in the literature.

Fig.3. Schematic diagram of the proposed approach

2.1 Preprocessing

The scanned character is first converted into binary image containing '0' and '1' pixels only. '0' indicates the character black pixels. Pre-processing techniques like thinning, slant correction and smoothing are applied. Extra rows and columns containing only '0's are removed from all four sides of the character. Finally the image is normalized to the standard size of 32x32 [2].

2.2 Feature extraction

We studied two types of features. The first is the pixel density features [4] based on zoning technique. The other is the GIST features [15]. A brief review of these features is given below.

2.2.1 Pixel density features

We extract the Pixel density [6] from the character using the zoning technique. The image is divided into 8x8 blocks, each of 4 × 4 grid size as shown in Fig.2. By dividing the sum of all black pixels present in a box with their total number, the pixel density for each box as follows

$$\lambda = \frac{1}{N}\sum_{k=1}^{n_b} 1 \qquad (1)$$

where N is total number of pixels in a box. n_b is the number of black (pattern) pixels in b^{th} box. Thus we have 64 features for each character.

2.2.2 Gist Features

Oliva and Torralba proposed GIST descriptor [5] to represent the spatial envelope of the scene. The name is acronym to *gist* of the scene. Since then it has been used for several image classification tasks. The GIST descriptor of an image computes the windowed 2D Gabor filter responses of an input image. The responses of Gabor filters encode the texture gradients that describe the local properties of the image. Averaging out these responses over larger spatial regions gives us a set of global image properties.

In the current work, we have computed GIST descriptor over 4x4 grid with 5 scales and 10 orientations. Thus the feature vector for each grayscale character image was of order 16 x (5+10) = 240.

2.3 Outlier detection

Supposing for each character there are 'n' possible feature values (64 or 240 in this work) and 'm' samples. Then a particular feature value gathered over all the samples forms a set. It is very rare that a feature value will be the same in all the samples. Due to preprocessing techniques like thinning or owing to handwriting styles or clarity, some of the features values collected in a feature set can turn out to be out of context. Such feature values, can be called as outliers from the statistics angle of view. We used the Grubb's test [1, 5] for outliers for filtering each feature set and studied it's effect on the recognition rate.

Grubbs' test is a statistical test used to detect outliers in a univariate data set with the assumption that it has normal distribution. It detects one outlier at a time. This outlier is excluded from the dataset and the test is iterated until no outliers are detected. For a significance level α, and P number of points in a feature set, a point is identified as outlier for the following condition

$$G > \frac{P-1}{\sqrt{P}}\sqrt{\frac{t^2_{\alpha/(2P),P-2}}{P-2+t^2_{\alpha/(2P),P-2}}} \qquad (2)$$

with $t^2_{\alpha/(2P),P-2}$ denoting the upper critical value of the *t*-distribution with $P-2$ degrees of freedom and a significance level of $\alpha/(2P)$.

All the outliers thus collected in a feature set, are replaced by the following two methods

1. Replace the outliers with the mean of the remaining values in the feature set.
2. Replace the outliers with randomly selected values from the remaining values in the feature set.

This replacement is carried out for each feature in a set (64 pixel density features set or 240 GIST feature set) separately.

It has to be noted that the outlier replacement technique is not overfitting the classifier. For only one feature value amongst 64 (or 240) feature values has been replaced occasionally as the case is. Also this replacement technique is applied on the training data only. If the testing data (which is always sure!) contains any outliers, the results will reflect

whether the classifier is properly trained or merely manipulated!

Grubb's test assumes that the given set has Normal distribution. In reality it is not so. To keep our goal in focus, we also handled another case - the case of assuming that the feature distribution has log-normal distribution. In such a case, it is valid to apply *log* to the distribution [1] and apply the same Grubb's test. Afterwards, all the values are applied to the transformation of *exp*() to bring back to their original distribution.

3. CLASSIFIER

We use Support Vector Machine (SVM) [12] as classifier in this study. The SVM produces a model (based on the training data) which predicts the target values of the test data given only the test data features. An SVM is defined for two-class classification. Given a training set of instance-label pairs (x_i, y_i), $i = 1, 2 \ldots l$ where $x_i \in R^n$ and $y \in \{1,-1\}^l$, the SVM require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{3}$$

Subject to $\quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \qquad \xi_i > 0$

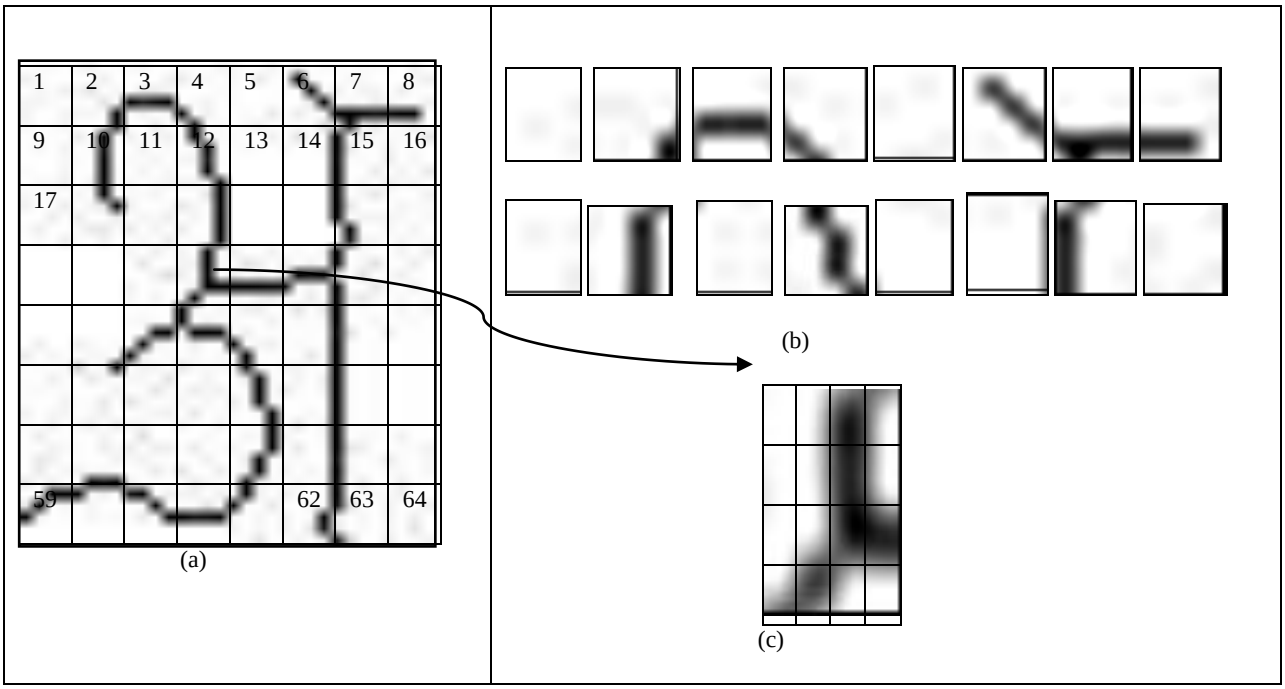


Fig 4. (a) Handwritten character 'अ' divded into 8x8 blocks. with feature indices (b) top 2 rows after zoning and (c) 4x4 grid placed on one block to compute the **Pixel** density feature.

Here the training vectors x_i are mapped into a higher dimensional space by the function ϕ. SVM finds the optimal hyper-plane which maximizes the distance, or more specifically the *margin*, between the nearest examples of both the classes. These nearest examples are called as support vectors (SVs). C > 0 is the penalty parameter of the error term.

Furthermore, $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is called the kernel function. We used the radial basis function (RBF) kernel in our work given by $K(x_i, x_j) \equiv e^{(-\gamma \|x_i - x_j\|^2)}, \gamma > 0$.

The SVM is implemented in MatLab [www.mathworks.com] using the LIBSVM software [3]. There are two parameters for an RBF kernel: C and γ. The parameter search for best C and γ is obtained by conducting a grid search and using 5-fold cross-validation. Various pairs of (C, γ) values are tried and the one with the best 5-fold cross-validation accuracy is picked. Trying exponentially growing sequences of C and γ is a practical method to identify good parameters (for C = 2^{-5}, 2^{-3},…, 2^{15}, and γ = 2^{-3}, 2^{-1},0,…, 2^{15}).

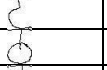
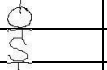
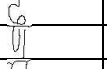
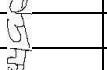
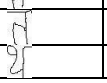
4. RESULTS AND DISCUSSIONS

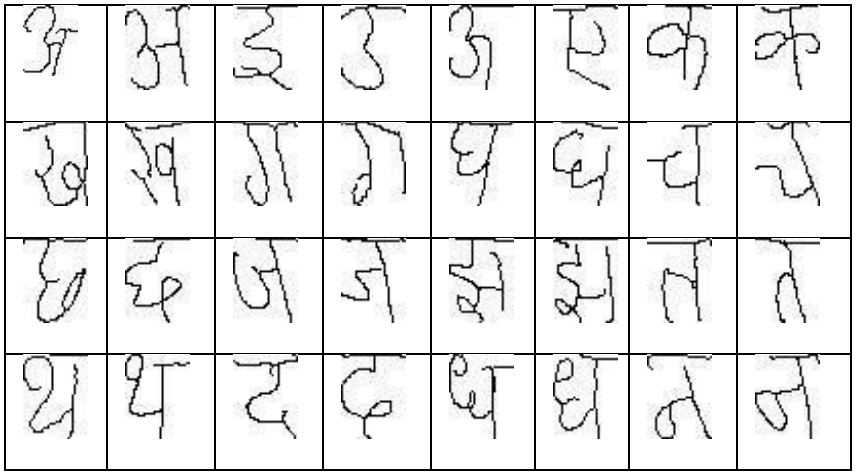
The proposed model is applied on two databases of handwritten Devanagari characters separately to see its effectiveness. The details of these databases are shown in Table 1. These databases can be obtained by mailing to the corresponding authors of references given in Table 1. These two databases were chosen amongst many owing to their special features. ISI database is the largest database available in this area and contains as much realistic samples as possible. The IIT Database has varying number of training and testing samples for each class. The details are given in Table 2. Such a database is typical case of imbalance datasets and demands robust classification techniques.

Table 1: Details of Database used

Database	Source	No of classes	Size of database
IIT Database	[7]	36	4713
ISI database	[9,10]	47	36172

Table 2: Details of IIT Database

Character	Training	Testing
	150	24
	100	25
	50	27
	170	39
	60	27
	175	29
	80	28
	180	33
	170	30
	140	24
	100	36
	140	26
	180	35
	75	20
	120	39
	80	30
	80	37
	25	7
	120	29
	80	31
	100	20
	100	39
	55	24
	40	16
	40	18
	170	36
	60	26
	140	25
	85	30
	80	33
	130	21
	70	24
	100	37
	40	15
	120	36
	100	32
Total	3705	1008



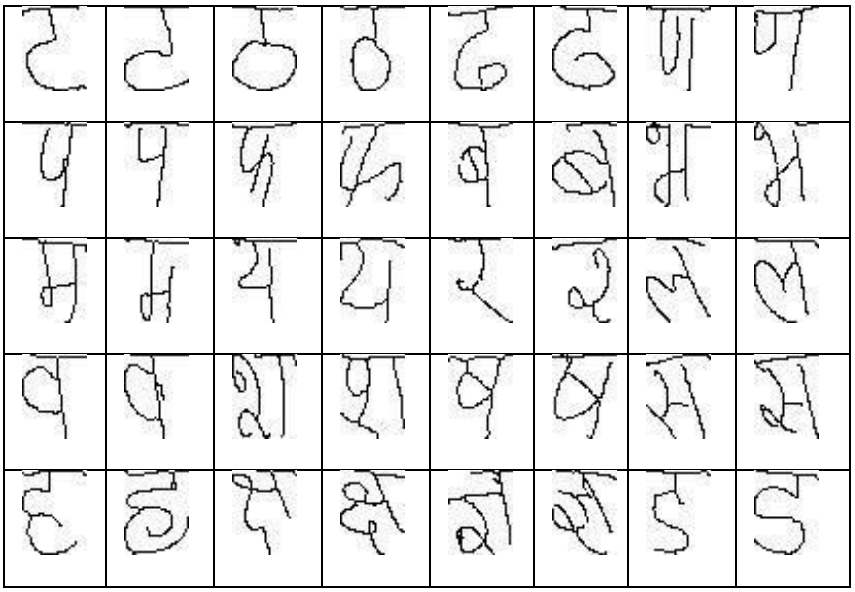


Fig. 6: Snap shot of preprocessed Devanagari Characters of IIT Database(courtesy [7])

Table3: Recognition rates on IIT Database using pixel density features and GIST features for normal distribution

Significance level	Without filtering	outlier filtering using method 1	outlier filtering using method 2	Without filtering	outlier filtering using method 1	outlier filtering using method 2
0.025	88.9	93.1	92.9	94.8	96.3	96.3
0.050	88.9	93.5	93.4	94.8	96.5	**96.5**
0.100	88.9	94.1	94.0	94.8	96.	96.6
0.200	88.9	94.6	94.4	94.8	96.9	97.1

Table4: Recognition rates on IIT Database using pixel density features and GIST features for log-normal distribution

Significance level	Without filtering	outlier filtering using method 1	outlier filtering using method 2	Without filtering	outlier filtering using method 1	outlier filtering using method 2
0.025	88.9	91.1	91.1	94.8	95.7	95.7
0.050	88.9	91.3	91.3	94.8	95.7	95.7
0.100	88.9	91.9	91.9	94.8	95.9	96.0
0.200	88.9	92.5	92.5	94.8	96.5	96.3

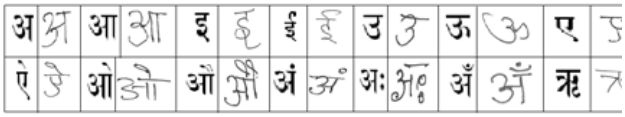
Table 5: Recognition rates on ISI Database using pixel density features and GIST features for normal distribution

Characters per class	Without outlier filtering	With outlier filtering using method 1	With outlier filtering using method 2	Without outlier filtering	With outlier filtering using method 1	With outlier filtering using method 2
500	76.0	80.7	80.6	84.9	86.7	86.7
200	67.4	73.8	73.6	80.0	82.8	**82.9**
100	61.9	70.3	70.1	75.2	79.0	78.9

Table 6: Recognition rates on ISI Database using pixel density features and GIST features for log-normal distribution

Characters per class	Without outlier filtering	With outlier filtering using method 1	With outlier filtering using method 2	Without outlier filtering	With outlier filtering using method 1	With outlier filtering using method 2
500	76.0	77.9	77.9	84.9	86.1	86.1
200	67.4	69.5	69.5	80.0	81.9	82
100	61.9	63.7	63.6	75.2	78.3	77.7

A snapshot of characters from the two databases is shown in Fig5 and Fig 6 respectively.



(a)

(b)

Figure 5: Samples of ISI database (a) Vowels (b) Consonants. Samples of printed characters are shown in the left side of the respective handwritten characters. (Courtesy [9])

4.2 Results on IIT Database

Results on IIT Database are shown in Tables 3 and 4. In each table, results pertaining to the two feature techniques discussed in 2.2 and the two methods of outlier replacements discussed in 2.3 have been summarized. Table 3 shows results with the assumption that features are having Normal distribution while Table 4 shows with log-Normal distribution. In many applications, the significance level α is chosen as 5% [13, 14]. We have verified our results on some more values of significance level. In all cases, the outlier filtering has shown significant improvement in the results. Also, the replacement values of outliers using two methods proposed was yielding nearly same. It is also worth observation that zoning features tend to be affected by outliers more than the processed features like the GIST, which seems to contain the inherent filtering technique. While there is an improvement of nearly 5% in the case of zoning features, there is only an improvement of 1% in the GIST features.

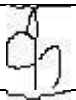
In comparison with the results in the literature, we achieved better performance than [7]. Also, [7] performed coarse classification into 5 sub-classes to address the class imbalance problem and small size of the database. Those considerations are not required in our methodology.

4.3 Results on ISI database

Results on ISI Database are shown in Tables 5 and 6. Table 5 shows results with the assumption that features are having Normal distribution while Table 6 shows with log-Normal distribution. Because the size of this database is very large, we used this database to study the effect of the size of the database on role of outliers. Three cases were studied. First case was with 100 characters per class, the second with 200 characters per class and the third with 500 characters per class. As observed from Table. 5, our proposed outlier detection and replacement has yielded an improvement of 6% to 10% in recognition rate in all the three cases. Performances using GIST features were very consistent and didn't yield improvement beyond 3%.

5-fold cross validation results have been reported to ensure that training set and testing set are completely chosen randomly. In all the cases, there has not been a single case of decrease in performance, for the testing cases may also have some outlier points. This also shows that out filtering technique is not over fitting a classifier. Some sample character images where outliers were detected for pixel density features is shown in Table.7. As pixel density features can be directly correlated visually, the respective feature index is also given in the table for easier understanding. These results show that filtering is effective

Table 7: Some samples where outliers were detected.

Database	Feature index (see Fig.4(a))	Actual sample	Ideal character
IIT	4		क
IIT	46		ह
IIT	1,30		प
ISI	4		अ
ISI	11		ऊ

A comparison of our best results with that in literature has been shown in Table 8. It can be seen that we are close the benchmark results, but with simpler approaches.

Table 8: Comparison with other works

Database	Size of database	Reference	Recognition rate
ISI	11270	[9]	80.36
ISI	9400	Proposed method	82.9%
IIT	4713	[7]	69.8%
IIT	4713	Proposed method	96.5%

4.4 Future work

Future work is investigation of extension of this filtering technique to other powerful features and classifiers. For instance, in [10] a 392 feature vector in conjunction with a Quadratic classifier was reporting 95.2% accuracy. We would like to study the effect of filtering in those areas also.

Also, the Grubb's test makes an assumption that the distribution is normal. But in practical cases, the distribution of the feature sets is not at all normal nor symmetrical. Outlier detection in such cases and better ways of replacing them for improvement of accuracy has to be investigated. Studies can also be made into effect of outliers in other character recognition of scripts.

5. CONCLUSIONS

A filtering technique in the feature extraction stage is proposed to improve the handwritten Devanagari character recognition. Using Grubb's test, outlier points are detected in each feature set collected for different character classes. By replacing them with some proximate points, a significant improvement in recognition rates was observed. This improvement is found to consistent irrespective of size of database, variation of database, type of features used and assumptions of the type of distributions of the features.

We hope this technique can serve other researchers in their works, substantially, on their features and classifiers for Devanagari character recognition. As many researchers in Indian languages are using zoning features, this filtering technique will be very useful for them.

6. ACKNOWLEDGMENTS

We are very grateful to the Department of Science and Technology, Government of India for supporting this work through a grant. We are extremely grateful to Dr. U. Pal, ISI, Kolkata for providing us the Devanagari character benchmark database.

.

7. REFERENCES

[1] V. Barnett and T. Lewis , "Outliers in Statistical Data", Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Chichester, 1994.

[2] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, Ching Y. Suen, "Character Recognition Systems: A Guide for students and Practioners", John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[3] 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.

[4] Vikas J Dongre and Vijay H Mankar, "A Review of Research on Devanagari Character Recognition", International Journal of Computer Applications, 12(2):8–15, December 2010.

[5] F.E. Grubbs, "Procedures for Detecting Outlying Observations in Samples", Technometrics, 11-1, pp.1--21; Feb., 1969

[6] M. Bokser, "Omnidocument technologies," Proc. IEEE, vol. 80, no. 7, pp. 1066–1078, Jul. 1992.

[7] M. Hanmandlu, O. V. Ramana Murthy, Vamsi Krishna Madasu, "Fuzzy Model based recognition of handwritten Hindi characters", Digital Image Computing Techniques and Applications, DICTA 3-5 Dec 2007, pp. 454-461.

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

[9] U. Pal, N. Sharma, T. Wakabayashi and F. Kimura, "Off-Line Handwritten Character Recognition of Devanagari Script", In Proceedings 9th International Conference on Document Analysis and Recognition. Pp. 496-500, Curitiba, Brazil, September 24-26, 2007.

[10] U. Pal, T. Wakabayashi and F. Kimura, "Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifiers", In Proc. 10th International Conference on Document Analysis and Recognition (ICDAR), pp.1111-1115, 2009

[11] N. Sharma, U. Pal, F. Kimura and S. Pal, "Recognition of Offline Handwritten Devanagari Characters using Quadratic Classifier", In Proc. Indian Conference on Computer Vision Graphics and Image Processing, pp-805-816, 2006

[12] V. Vapnik, "The Nature of Statistical Learning Theory", Springer Verlag, 1995.

[13] Stigler S, "Fisher and the 5% level". Chance, 2008, 21 (4): 12.

[14] Fisher R. A., Statistical Methods for Research Workers (first ed.). Edinburgh: Oliver & Boyd, 1925

.