

# Transform Domain Methods For Devanagiri Character Recognition

Deepika Rajurkar

Prof. AlwinAnuse

## Abstract

*Rapidly growing computational power enables the implementation of Devnagri Character Recognition methodologies. Digital document processing is gaining popularity for application to office and library automation, bank and postal services, publishing houses and communication technology. Character recognition system consists of three steps: (i) pre-processing (ii) feature extraction and (iii) classification. Features extracted, from the characters using transform domain methods are, Fourier Transform and Wavelet Transform. Three types of classifiers i.e Feed- forward Backpropagation, Generalized Regression Neural Network and K-Nearest Neighbor method are implemented in this paper. Results in terms of recognition rate using combinations of feature extraction methods and classifiers are compared and analysed.*

## 1. Introduction

Artificial Intelligence, giving machines the human like abilities, has remained one of the most challenging areas in Electronic sciences in last few decades. Giving machine the power to see, interpret and the ability to read text is one of the major tasks of AI. A good text recognizer has many commercial and practical applications, e.g. from searching data in scanned book to automation of any organization, like post office, which involve manual task of interpreting text. Devnagri script consists of 11 vowels and 33 consonants. In recognition of Devnagri characters recognition of different handwritten characters are done. Features are extracted from different handwritten and then recognition is done on the basis of extracted features. In Devnagri script some of the characters are similar and therefore recognition of these characters is difficult. Extensive research works on recognition of handwritten characters, have been carried out since the last few decades because it is a classical problem of pattern recognition and it has enormous application potentials. A wide variety of

algorithms / schemes exist for recognition of isolated handwritten characters and each of these has their own merits and demerits. The existing character recognition schemes differ from each other mainly with respect to pre-processing steps used, feature selection / extraction method and classifier used. Combining various pre-processing steps, feature extraction methods, and different classifiers high recognition rate can be achieved. Here, results, in terms of recognition rate, of different pre processing steps and feature extraction methods are compared and discussed [1].

## 2. Pre-processing

### 2.12-D Median-filtering

Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges. It performs median filtering of the matrix A in two dimensions. Each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image [2].

### 2.2 Binarization

Global image threshold using Otsu's method. It computes a global threshold (level) that can be used to convert an intensity image to a binary image [6].

### 2.3 Morphological operations

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image. The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an

image, while erosion removes pixels on object boundaries.[6].

## 2.4 Cropping

All the characters are uniformly sized and cropped so that further operations can be done only on concern character pixels leaving extra pixels aside.

## 2.5 Thinning

Thinning thins objects to lines. It removes pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary.

## 2.6 Segmentation

In most of the existing systems recognition accuracy is heavily dependent on the quality of the input document. In handwritten text adjacent characters tend to be touched or overlapped. Therefore it is essential to segment a given string correctly into its character component[2].

## 3. Feature Extraction

### 3.1 Fourier Descriptors Algorithm

The purpose of the algorithm is to extract the information of the boundary (contour) of a segmented character and present it in a more compact form. There are various methods of boundary tracing. The “four-neighbor” adjacent method is adopted in this system [3].

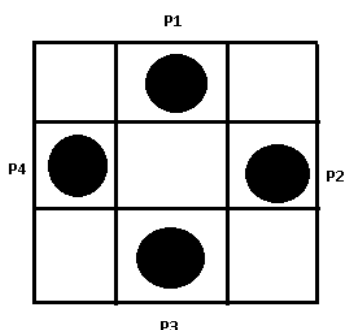


Figure. 1. Four-neighbor searching sequence

Algorithm:

- i. Remove noise by applying filter.
- ii. Convert gray scale image into binary image.
- iii. Remove noise (eg. Spikes) by applying erode and dilate morphological operations respectively.

- iv. To bring uniformity among the numerals, fit a bounding box to the character image, crop, and resize it to a size of 100x 100 pixels.
- v. Apply thinning to the character image.
- vi. Segment the character image into 25 blocks of size 4 x 4 size.
- vii. Extract the boundary of the character image and resample the boundary in order to obtain a uniform resampling along the running arc length of the boundary.
- viii. Represent the boundary in the complex plane where the column-coordinate is the real part and the row-coordinate the imaginary part.
- ix. Compute Fourier transform and obtain the invariant 32 dimensional Fourier descriptors.
- x. Calculate magnitude and phase of these 32 fourier descriptors.
- xi. Nullify the dc value of fourier descriptors as it contains very less information.
- xii. Calculate standard deviation of 31 phase values.
- xiii. Combining 31 magnitude values and 1 standard deviation value of 31 phase values, total features are 32.
- xiv. Normalize these 32 feature values and apply these to different classifiers to classify the characters.

### 3.2 Discrete Wavelets Algorithm

- i. Remove noise by applying filter.
- ii. Convert gray scale image into binary image.
- iii. Remove noise (eg. Spikes) by applying erode and dilate morphological operations respectively.
- iv. To bring uniformity among the numerals, fit a bounding box to the character image, crop, and resize it to a size of 100x 100 pixels.
- v. Apply thinning to the character image.
- vi. Segment the character image into 25 blocks of size 4 x 4 size.
- vii. Apply daubechies wavelet transform on the segmented image and calculate discrete wavelets for the segmented image.
- viii. Normalize these feature values and apply these to different classifiers to classify the characters.

### 3.3 Fusion (Fourier Descriptors + Discrete Wavelets)

- i. In case of fusion, combination of both the features i.e. Fourier descriptors and discrete wavelets, are used as extracted features.
- ii. These features are then normalized before applying to different classifiers to classify the characters.

## 4. Classification

### 4.1 Feed-forward Backpropagation Algorithm

1. Initialize all  $W_{ij}$ 's to small random values with  $W_{ij}$  being the value of the connection weight between unit  $j$  and unit  $i$  in the layer below.
2. Present the input vectors i.e. magnitude and phase. Specify the desired outputs. If the net is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from.
3. Calculate the outputs  $y_j$  of all the nodes using the present value of  $W_{ij}$  where  $W_{ij}$  is the value of connection weight between unit  $j$  and the unit  $i$  in the layer below:

$$y_j = \frac{1}{1 + \exp(-\sum y_i W_{ij})} \quad \dots \text{Eqn(1)}$$

This particular nonlinear function is called a sigmoid function.

4. Adjust weights by:

$$W_{ij}(n+1) = W_{ij}(n) + \alpha \delta_j y_i + \xi (W_{ij}(n) - W_{ij}(n-1)) \quad \dots \text{Eqn (2)}$$

where  $0 < \xi < 1$

where  $(n+1)$ ,  $(n)$  and  $(n-1)$  index next, present and previous, respectively. The parameter  $\alpha$  is a learning rate similar to step size in gradient search algorithms, and  $\xi$  is a constant between 0 and 1 which determines the effect of past weight changes on the current direction of movement in weight space.  $\delta_j$  is an error term for node  $j$ . If node  $j$  is an output node,  $d_j$  and  $y_j$  stand for respectively, the desired and actual value of a node, then

$$\delta_j = (d_j - y_j) y_j (1 - y_j) \quad \dots \text{Eqn (3)}$$

If node  $j$  is an internal hidden node, then :

$$\delta_j = y_j (1 - y_j) \sum \delta_k W_{jk} \quad \dots \text{Eqn (4)}$$

where  $k$  is over all nodes in the layer above node  $j$ .

5. Present another input and go back to step (2). All the training inputs are presented cyclically until weights stabilize (converge).

Following are specifications of the classifier used for training:

- i. Learning rate=0.05
- ii. Epochs=50000
- iii. Training parameter=50
- iv. Goal=1e-1
- v. 3 layers: 1 input layer, 1 hidden layer, 1 output layer.
- vi. Hidden layer uses function tangent sigmoid and output layer pure linear.
- vii. Levenberg-Marquardt backpropagation recognition engine is used.

### 4.2 Generalized Regression Neural Network Algorithm

1. Calculate the statistically most likely value of each output from a given set of inputs, all taken from the training set. This most likely value is the conditional mean.
2. The algorithm requires the joint probability density function of the inputs and outputs, each treated as random variables. The calculation is done for each output individually. The joint probability density function of vector  $x$  inputs and  $y$  output may be written as

$$p(x|y) = \frac{1}{(2\pi\sigma^2)^{N/2} N_{\text{train}}^{1/2}} \sum_{j=1}^{N_{\text{train}}} \left( \exp\left(-\frac{d_j}{2\sigma^2}\right) \exp\left(-\frac{(y-y_j)^2}{2\sigma^2}\right) \right) \quad \dots \text{Eqn(5)}$$

$$d_j = \sqrt{\sum_{j=1}^N (x - x_j)^2} \quad \dots \text{Eqn(6)}$$

where,  $N$  is the number of input nodes,  $N_{\text{train}}$  is total number of training sets,  $\sigma$  is the width parameter subject to certain constraints, and  $d_j$  is the Euclidean distance between an input and the mean of those inputs.

3. The statistically most likely value for the random variable  $y$  (output) given (the vector of random input variables) is

estimated substituting Eqn(5) and Eqn(6) into

$$y(\bar{X}) = \frac{\int_{-\infty}^{\infty} y(p d(\bar{X} | y)) dy}{\int_{-\infty}^{\infty} p d(\bar{X} | y) dy} \dots \text{Eqn(7)}$$

- The algorithm uses each of the training vectors as the centre of a Gaussian function defined as  $\exp(-D_j^2/2\sigma^2)$ . The larger the number of training samples, the larger is the amount of memory required and the longer is the time for the calculation.

### 4.3 K-Nearest Neighbor Algorithm

- Form the training dataset P.
- Given a query instance to be classified x.
- Let  $x_1, x_2, \dots, x_k$  denote the k instances from training dataset that are nearest to x
- Return the class that represents the maximum of the k instances.
- Distance can be calculated from different methods i.e Euclidean, Cityblock, Hamming etc.
- By default it calculates distance using Euclidean method.
- Values of k varies in odd sequence i.e  $k=1,3,5, \dots$ . Default value is 1.

## 5. Testing and results

### 5.1 Database

Database is taken from Indian Statistical Institute, Kolkata. Database consists of 33 consonants. 10 and 20 samples are taken of each character to form the sets of database. In this way three types of database sets are formed:

- Training sample = 660; Testing sample = 330
- Training sample = 660 ; Testing sample =330
- Training sample = 330 ; Testing sample = 660

### 5.2 Results of Pre-processing

Original images

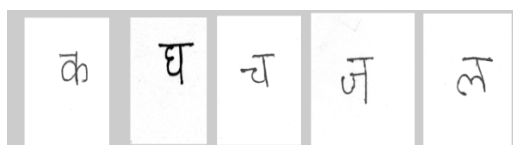


Figure 2. Original Images

Median filtered images



Figure 3. Median Filtered Images

Dilated and Eroded images



Figure 4. Dilated and Eroded Images

Cropped images



Figure 5. Cropped Images

Thinned images



Figure 6. Thinned Images

Segmented images

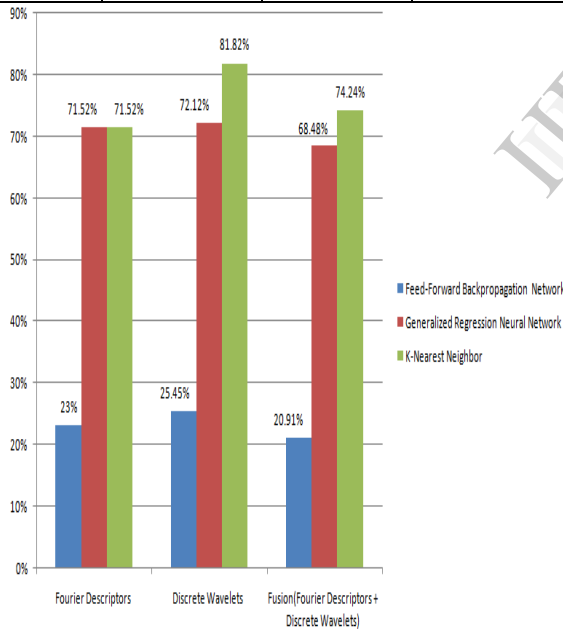


Figure 7. Segmented Images

### 5.3 Comparison of recognition rate

**Table 1- Comparison of recognition rate using database (a)**

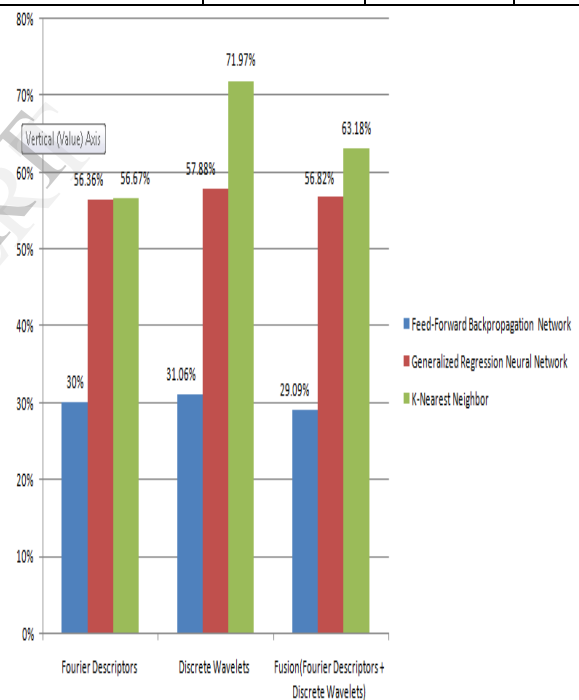
FEATURE EXTRACTION METHODS TYPES OF CLASSIFIERS	RECOGNITION RATE		
	Fourier Descriptors	Discrete Wavelets	Fusion (Fourier Descriptor + Discrete Wavelets)
Feed-Forward Backpropagation Network	23%	25.4545%	20.9091%
Generalized Regression Neural Network	71.5152%	72.1212%	68.4848%
K-Nearest Neighbour	71.5152%	<b>81.8182%</b>	74.2424%



**Graph 1-Graphical Representation of Comparison of Recognition Rates using Database (a)**

**Table 2- Comparison of recognition rate using database (b)**

FEATURE EXTRACTION METHODS TYPES OF CLASSIFIERS	RECOGNITION RATE		
	Fourier Descriptors	Discrete Wavelets	Fusion (Fourier Descriptor + Discrete Wavelets)
Feed-Forward Backpropagation Network	30%	31.0606%	29.0909%
Generalized Regression Neural Network	56.3636%	57.8788%	56.8182%
K-Nearest Neighbour	56.6667%	<b>71.9697%</b>	63.1818%

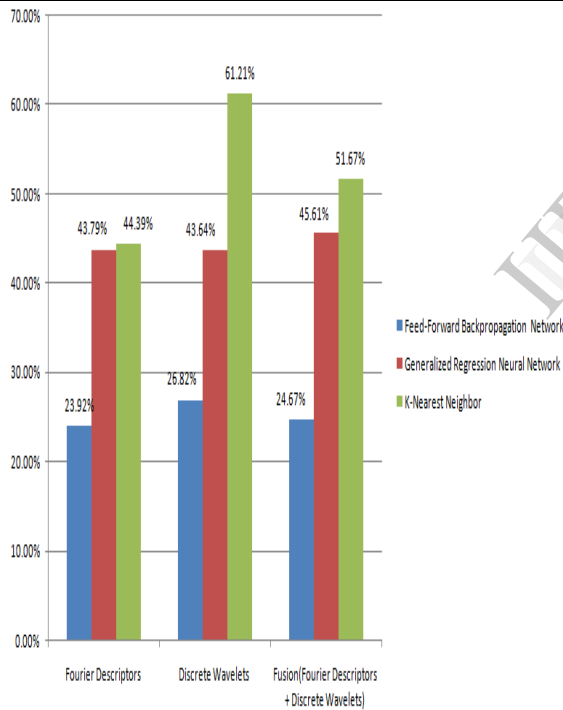


**Graph 2-Graphical Representation of Comparison of Recognition Rates using Database (b)**

**Table 3- Comparison of recognition rate using database (c)**

instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes i.e. commonly mislabeling one as another. Here, confusion matrices are shown of system consists of Discrete Wavelets and KNN of all the 3 database.

FEATURE EXTRACTION METHODS / TYPES OF CLASSIFIERS	RECOGNITION RATE		
	Fourier Descriptors	Discrete Wavelets	Fusion (Fourier Descriptor + Discrete Wavelets)
Feed-Forward Backpropagation Network	23.9191%	26.8182%	24.6697%
Generalized Regression Neural Network	43.7879%	43.6364%	45.6061%
K-Nearest Neighbour	44.3939%	<b>61.2121%</b>	51.6667%



Graph 3-Graphical Representation of Comparison of Recognition Rates using Database (c).

5.4 Confusion Matrix

In the field of machine learning, a confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each column of the matrix represents the instances in a predicted class, while each row represents the

Table 4-Confusion Matrix of Discrete Wavelets using KNN with Database (a)

	क	ख	ग	घ	च	छ	ज	झ	ञ	ट	ठ	ड	ढ	ण	त	थ	द	ध	न	प	फ	भ	म	य	र	ल	व	श	ष	स	ह	ऋ
क	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	
ख	0	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
ग	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	
घ	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	
च	0	0	0	0	8	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
छ	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ज	0	0	0	0	0	0	7	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
झ	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
ञ	0	0	0	0	0	0	1	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
ट	0	0	0	0	0	0	0	0	0	7	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
ठ	0	0	0	0	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ड	0	0	0	0	0	0	0	0	0	1	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ढ	0	0	0	0	0	0	0	0	0	0	1	1	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ण	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
त	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
थ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	
द	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
ध	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	
न	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	2	0	0	0	0	0	0	
प	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	
फ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	1	1	0	0	0	0	0	0	
भ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	1	0	0	1	0	0	0	0	
म	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	7	0	0	0	0	0	0	0	0	
य	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	9	0	0	0	0	0	0	0	0	
र	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	
ल	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	
व	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	8	0	0	0	0	0	
श	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	
ष	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	
स	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	8	0	0	
ह	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	
ऋ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0

Table 5-Confusion Matrix of Discrete Wavelets using KNN with Database (b)



- If Training samples are more than testing samples than recognition rate is high.
- If Discrete Wavelets are used as Features i.e Discrete Wavelet Transform is used as Feature Extraction method than recognition rate is comparatively more.
- If K-Nearest Neighbor is used as Classifier than recognition rate is high.

**MATLAB**", publication: Dorling Kindersely (India) Private Limited, edition:2006.

[7] Database: ISI, Kolkata

## References

[1] Yuk Ying Chung, Man To Wong; **"Handwritten Character Recognition By Fourier Descriptors And Neural Network"**; IEEE TENCON on Speech and Image Technologies for Computing and Telecommunications, 1997, Australia, page(s):391-394.

[2] Anshul Gupta, Manisha Srivastava, Chitralkha Mahanta; **"Offline Handwritten Character Recognition Using Neural Network"**; IEEE International Conference on Computer Applications And Industrial Electronics(ICCAIE), 4-7 Dec. 2011, Penang, page(s):102-107.

[3] Bhattacharya U., Vajda S., Mallick A., Chaudhuri B.B., Belaid A., **"On The Choice Of Training Set, Architecture and Combination Rule Of Multiple MLP Classifiers For Multiresolution Recognition Of Handwritten Characters"** IEEE Transactions On frontiers in Handwriting Recognition, Kolkata, India, page(s):419-424, 2004.

[4] S.B Patil, G.R Sinha, V.S Patil, **"Recognition of Devnagri Characters"** 2<sup>nd</sup> International Conference on emerging Applications of Information technology (EAIT), India, page(s):185-189, 2011.

[5] Arora S., Bhattacharjee D., Nasipuri M.; **"Combining Multiple Feature Extraction Techniques for Handwritten Devnagri Character Recognition"**; IEEE Region 10 and the Third International Conference on Industrial and Information Systems, 8-10 Dec. 2008, Kharagpur, page(s):1-6.

[6] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, **"Digital Image Processing Using**