

# Design of an Effective Preprocessing Approach for Offline Handwritten Images

Dimple Bhasin  
Research Scholar  
Chandigarh University, India

Gulshan Goyal  
Associate Professor  
Chandigarh University, India

Maitreyee Dutta, Ph.D  
Associate Professor & Head  
NITTR Chandigarh, India

## ABSTRACT

Handwritten pattern recognition involves conversion of scanned images of handwritten patterns into a computer processable form. To recognize handwritten patterns is an easy and trivial task for human beings, but for a machine it is a cumbersome and a difficult task due to high variations in the shape of characters and writing style. Although complicated to train, yet machines can be useful in providing solution to the recognition problem. They save time and money and eliminate the requirement of execution of repetitive tasks by humans. In order to have better recognition the image should be properly pre-processed. Pre-processing reduces and eliminates noise and irregularities. The present paper focuses on different approaches to pre-processing and an insight to general methodology for the recognition process.

## General Terms

Pattern Recognition

## Keywords

Handwritten Pattern Recognition, Pre-Processing, Filters, Thinning, Artificial Neural Networks, Feature Extraction and Recognition.

## 1. INTRODUCTION

Pattern recognition is a field concerned with machine recognition of meaningful irregularities in noisy and complex environment [2]. Pattern recognition aims to classify data (patterns) based on either prior knowledge or statistical information extracted from the patterns [5]. Handwritten patterns recognition is of two types: online and offline as shown in Fig. 1.

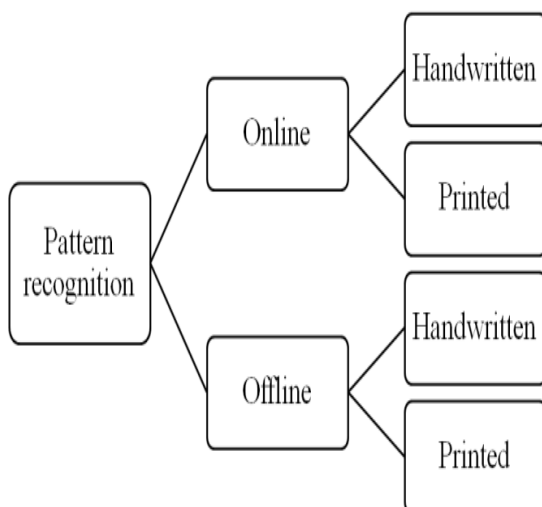


Fig. 1: Classification of Pattern Recognition

Online recognition system includes several pre-defined pre-processing steps to reduce variations and at the same time preserve information for recognition. But for offline recognition there are no pre-defined sequences of pre-processing steps [3]. Offline recognition is inherently difficult because of high variability of writing styles. Therefore, in offline recognition systems the pre-processing steps are defined with the basic objective of noise removal and elimination of irregularities.

## 1.1 Image analysis

An image may be defined as a two-dimensional function,  $f(x,y)$  where  $x$  and  $y$  are spatial co-ordinates and the amplitude  $f$  at any pair of co-ordinates  $(x,y)$  is called intensity or gray-level of image at that point [9]. If  $x$ ,  $y$  and  $f$  are finite, the image is called a digital image. Basically there are 4 types of images:

1. Binary images
2. Gray-level images
3. RGB or true color images
4. Indexed images

An image can have different formats like jpeg, tiff, bmp, png etc. An image can be analyzed using: Histograms, Resolution, size, and number of connected component etc. as shown in Fig. 2.

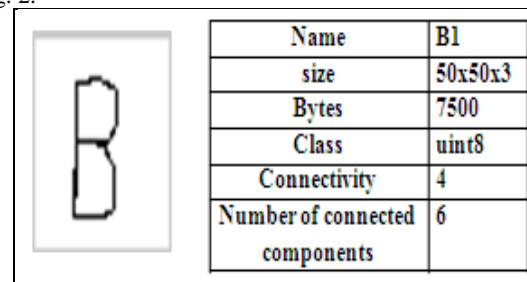


Fig. 2: Input Image and its Properties Obtained Using 'whos' and 'bwconncomp' Command.

## 2. LITERATURE REVIEW

Offline Handwritten pattern recognition has been one of the fascinating and challenging research area in the field of image processing. Tremendous research has already been done and it is still being done to get a more efficient and user friendly approach for the problem. A brief review of various research papers is laid down in following paragraphs.

In [1], an overview of existing character recognition techniques, historical evolution of character recognition systems and current status of character recognition system is presented. The paper emphasizes on offline handwriting recognition problem. An efficient approach to pre-processing is presented comprising of noise reduction, normalization, and

compression. This approach needs to be applied carefully in order to avoid unexpected distortions or loss of data.

In [2], a system for classification and recognition of printed Hindi vowels and consonants using artificial neural network is presented. The system consists of acquiring an image, step by step pre-processing, and then grouping of characters followed by recognition of characters. The core part of system is to extract significant features in order to divide characters into sub-group. The extraction of features directly depends upon pre-processing steps comprising of binarization, noise elimination, and size normalization.

In [3], a system for recognition of unconstrained offline handwritten texts using Hybrid Markov model/ Artificial Neural network is presented. The paper also presents an efficient technique for pre-processing. The pre-processing steps include: Image cleaning, slope removal, slant removal, size normalization. All these steps involve using different types of Multi layer perceptrons. An efficient preprocessing approach removes the irregularities and distortions in image and thereby, improves recognition.

In [4], recognition of handwritten Devnagari scripts is presented. The basic approach for recognition consist of image acquisition, separating the lines, words and characters step by step using segmentation, and recognition of characters using neural network. A good insight of the pre-processing steps is presented. After the recognition the character is replaced by standard fonts. The size of the dataset imposes a large impact of recognition rate.

### 3. RECOGNITION PROCESS

The whole recognition process for offline handwritten patterns consists of following stages as shown in Fig. 3:

- 1) Image acquisition
- 2) Pre-processing
- 3) Feature Extraction
- 4) Classification
- 5) Post-Processing

#### 3.1 Image Acquisition

The images used are obtained from www.mathwork.com. The available dataset comprises of 650 images of 26 English alphabets out of which 130 images have been sorted. These 130 images contain 5 samples of different handwritings for each alphabet. Each image is of size 50x50 having bmp format. A sample of dataset is given in Fig. 4.

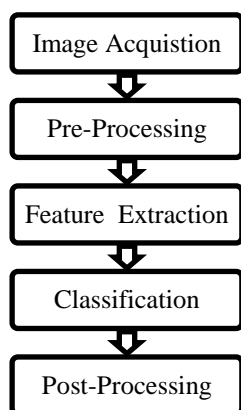


Fig. 3: Schematic diagram of the Recognition process

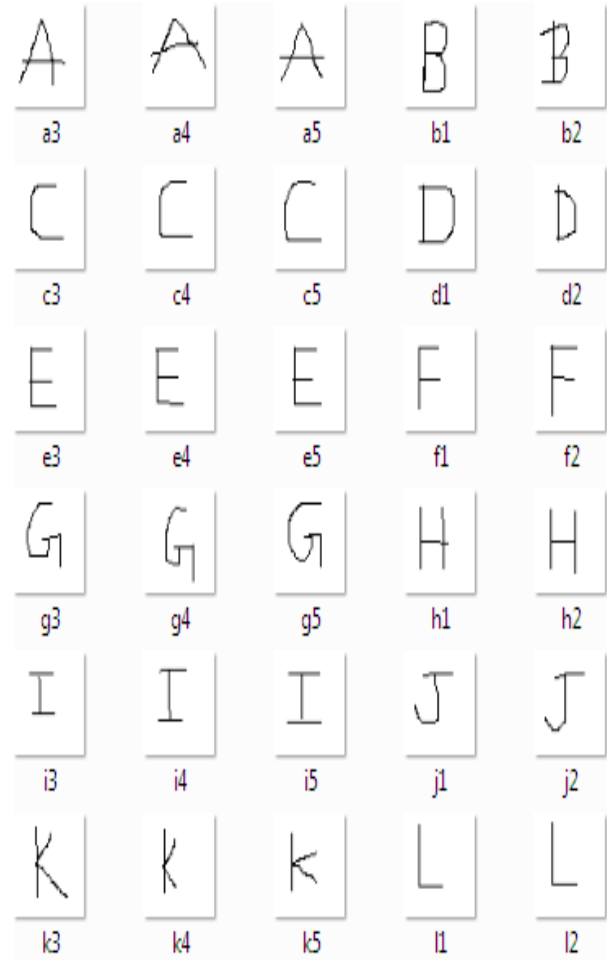


Fig. 4: Dataset of Handwritten Input images.

#### 3.2 Pre-processing

Pre-processing consist of different steps to mould the input image into a form suitable for processing. The main objectives of pre-processing are:

- 1) Noise reduction
- 2) Normalization of the data
- 3) Compression in the amount of information to be retained.

The pre-processing steps handle the necessary computations and operations on patterns before feeding them as input to neural networks. For the efficient recognition of handwritten patterns, the defined sequence of pre-processing steps needs to produce a properly furnished image free of any irregularities. It aims to produce data that are easy for the system to operate accurately. The presented Pre-processing approach consists of:

1. Image Cropping
2. Noise reduction
3. Image Binarization
4. Image Complement
5. Normalization
6. Thinning

These steps involved in pre-processing are shown in Fig. 5.

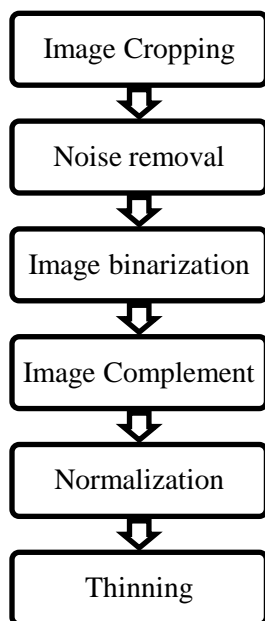


Fig. 5: Design of Pre-processing Approach for Offline Handwritten Images

### 3.2.1 Image cropping

It involves removal of surplus pixels and blank spaces from the image. It can be automated or done manually. In the present work, cropping is done manually using ‘imcrop’ command present in MATLAB as shown in Fig. 6. noise (Gaussian), Local variance (Gaussian with intensity-

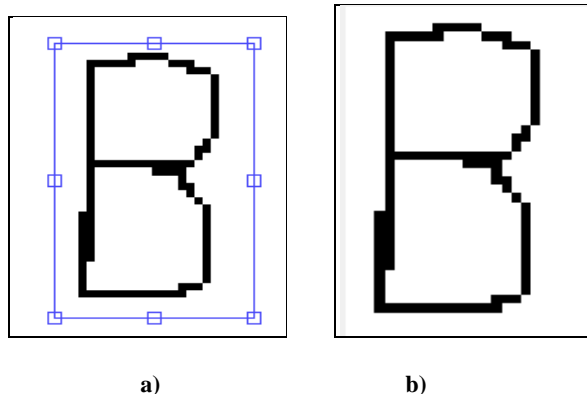


Fig. 6: a) Original Image b) Cropped Image obtained using ‘imcrop’ command.

### 3.2.2 Noise removal

The noise present in images causes disconnected line segments, gaps in lines and filled loops. Image noise represents unwanted or undesired information that can occur during the image capture, transmission, processing or acquisition, and may be dependent or independent of the image content. It is necessary to eliminate these imperfections [1].

#### 3.2.2.1 Filtering

It aims to remove noise and spurious points, introduced by uneven writing surface and poor sampling rate of data acquisition device [5]. Common types of noise are: White dependent variance), Salt and pepper Speckle. Speckle

degrades the images and thereby reduces the ability for human observer to discriminate the fine details in the image. Main types of filtering methods are shown in Fig.7.

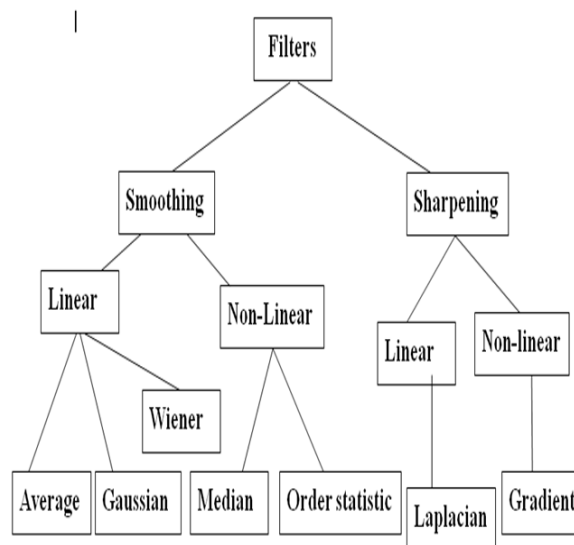


Fig.7: Different types of filters

## 1. SMOOTHING FILTERS

Smoothing filters are used for noise reduction and blurring operations.

### 1.1 Smoothing linear filter

These filters are used primarily for blurring and noise reduction. These kinds of filters are called averaging filters or low pass filters. Gaussian filter and wiener filter are types of smoothing spatial linear filters [6].

### 1.2 Smoothing non-linear filters

Nonlinear filter produce results that vary in a non-intuitive manner. Median filter and order statistics filter are two types of non-linear filters.

## 2. SHARPENING FILTERS

The main aim in image sharpening is to highlight fine detail in the image, or to enhance detail that has been blurred (perhaps due to noise or other effects, such as motion). Sharpening is used to enhance the image details by adding the edges to the input images [6].

### 2.1 Sharpening linear filters

This filter is a high pass spatial filter. These filters remove low frequency components and enhance high frequency components. Laplacian filters are a type of sharpening linear filters.

### 2.2 Sharpening non-linear filters

Sharpening used to enhance line structures or other details in an image. Gradient used to enhance prominent edges.

A comparison of smoothing and sharpening filters is given in Table 1.

**Table 1: Comparison of Different Filters.**

	Linear	Non-Linear
<b>Smoothing Filters</b>	Noise reduction; Blurring	Removes noise in an intuitive manner
	<b>Example:</b>  1. Gaussian Filters 2. Wiener Filter	<b>Example:</b>  1. Median Filters 2. Order Statistics Filter(max, min, median filters)
<b>Sharpening Filter</b>	Removes low frequency components;  Enhance high frequency components	Enhance line structures or other details in image.
	<b>Example:</b> Laplacian Filters	<b>Example:</b> Gradient Filter

**3.2.2.2 Performance evaluation of filters**

Different filters that can be applied to the image are evaluated in terms of:

1. Mean Square error (MSE)
2. Signal to noise ratio (SNR)
3. Peak signal to noise ratio (PSNR)
4. Root mean square error (RMSE)

**1. Mean Square Error (MSE):**

It refers to some of average or sum of square of the error between two images. MSE of Image can be calculated as [10]:

$$MSE = I/MN \sum_{i=0}^{m-1} [I(i, j) - K(i, j)]^2$$

Where I(i,j) = Noisy Image  
K(i,j) = Original Image  
M,N = Size of image

**2. Signal to Noise Ratio (SNR) :**

The mean square error is expressed in terms of a signal to noise ratio (SNR). SNR of image can be calculated as [10]:

$$SNR = 10 \log \sigma^2 / \sigma^2_e$$

Where  $\sigma^2$  = Variance of the desired image.  
 $\sigma^2_e$  = Average variance.

**3. Peak Signal to Noise ratio (PSNR) :**

Peak Signal to noise ratio is defined as the ratio between signal variance and reconstruction error variance. PSNR of Image can be calculated as [10]:

$$PSNR = 10 \log_{10} (Size/MSE)$$

**4. Root Mean Square error**

The Root Mean Square Error is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modeled. RMSE of Image can be calculated as [11]:

$$RMSE = \sqrt{I/MN \sum_{i=0}^{m-1} [I(i, j) - K(i, j)]}$$

Where I(i,j) = Noisy Image  
K(i,j) = Original Image  
M,N = Size of image

Smaller the value of MSE, and larger the value of SNR and PSNR, lesser will be the noise and higher the quality of image. A comparative analysis of different parameters is given in Table 2.

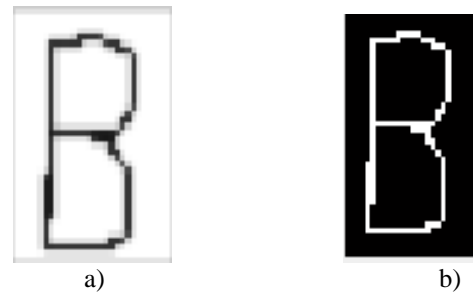
According to Table 2, Gaussian filters possess lowest value of MSE and highest value of SNR and PSNR. It reduces noise and improves the quality of images effectively. Hence, Gaussian filter is used for filtering.

**3.2.3 Image binarization**

Binarization involves conversion of a gray scale image into a binary image where each pixel taking a value of 0 or 1.

**3.2.4 Image complement**

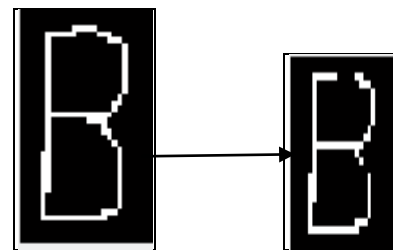
Image complement involves conversion of image into its negative i.e. 0 is converted to 1 and vice-versa. This is done since pre-processing steps involve thinning. Thinning can be done only on the complement of such images. An example of image complement is shown in Fig. 8.



**Fig. 8: a) Filtered Image b) Image Complement**

**3.2.5 Normalization**

It aims to obtain a standardized data. Size normalization adjusts the character size to a certain standard. In the present work binarized and complemented image is normalized to a size of 30x30 as shown in Fig. 9.



**Fig. 9: Complemented image resized to 30x30**

Table 2: Comparative analysis of different performance evaluation parameters

Input Image	Filter	Filtered image	PSNR	SNR	MSE	RMSE
	Average Filter		12.24	10.53	3913.06	62.55
	Gaussian Filter		21.41	10.50	473.54	21.76
	Wiener Filter		20.05	10.06	648.48	25.47
	Order Statistics Filter		5.29	Inf	1939.13	139.28

### 3.2.6 Thinning

Thinning is a morphological operation that removes selected foreground pixels from binary images and is particularly used for skeletonization [8]. It removes the points in such a way that only the skeleton of a branch pattern remains as shown in Fig. 11.

Thinning can remove irregularities in letters and in turn, makes the recognition algorithm simpler because they only have to operate on a character stroke, which is only one pixel wide. It also reduces the memory space required for storing the information about the input characters and also reduces the processing time too [4] [2].

In the present work, thinning is done using fast parallel thinning algorithms namely Zhang Suen algorithm and Guo Hall algorithm.



Fig. 10: (a) Examples of images and (b) their thinned versions [12]

### 3.2.6.1 Zhang Suen Algorithm

A binary digitized image is defined by a 3x3 window where each pixel is either 1 or 0. The pattern consists of those pixels that have value 1. Iterative transformation are applied to the window point by point according to values of a small set of neighboring points as shown in Fig. 13.

In order to preserve the connectivity of Skelton, each iteration is divided in to two sub-iterations. In the first sub iteration, the contour point p1 is deleted from the digital pattern if it satisfies following conditions:

- (a)  $2 \leq B(p1) \leq 6$
- (b)  $A(p1) = 1$
- (c)  $P2 * P4 * P6 = 0$
- (d)  $P4 * P6 * P8 = 0$

Where A (p1) is the number of 01 patterns in the ordered of p2, p3, p4...p8, p9 that are the eight neighbors of p1 (Fig. 12 above), and B(p1) is the non-zero neighbors of p1, that is  $B(p1) = p2+p3+...+p9$ .

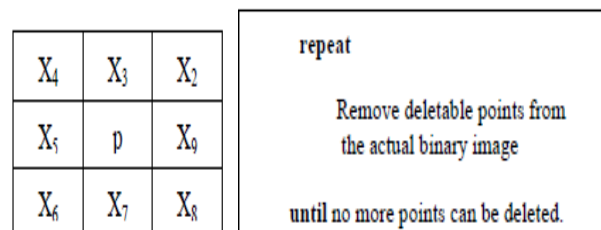


Fig. 11: a) Neighbors of a pixels p b) A thinning algorithm

P <sub>9</sub> (i-1,j-1)	P <sub>2</sub> (i-1,j)	P <sub>3</sub> (i-1,j+1)
P <sub>8</sub> (i,j-1)	P <sub>1</sub> (i,j)	P <sub>4</sub> (i,j+1)
P <sub>7</sub> (i+1,j-1)	P <sub>6</sub> (i+1,j)	P <sub>5</sub> (i+1,j+1)

Fig. 12: Designation of 9 pixels in 3X3 window: IT

If any condition is not satisfied then p1 is not deleted from the picture. In the second subiteration, only condition (c) and (d) are changed as follows:

$$(c') p2 * p4 * p8 = 0$$

$$(d') p2 * p6 * p8 = 0$$

and the rest remain the same.

By condition (c) and (d) of the first subiteration it will be shown that the first subiteration removes only the south-east boundary points and the north-west corner points which do not belong to an ideal skelton. By condition (a), the end-points of a skelton line are preserved. Also, condition (b), prevents the deletion of those points that lie between the end-point of skelton line. The iterations continue until no more points can be removed [8].

### 3.2.6.2 Guo-Hall algorithm

Let C (P) be the number of distinct 8-connected components of 1's in Ps 8-neighborhood.

$$N(P) = \text{Min}(N1(P), N2(P))$$

$$N1(P) = (p1vp2)+(p3vp4)+(p5vp6)+(p7vp8)$$

$$N2(P) = (p2vp3)+(p4vp5)+(p6vp7)+(p8vp1)$$

An edge point will be deleted if it satisfies:

a) C(P)=1;

b) 2<=N(P)<=3;

c) Apply one of the following:

1. (P2vP3vP5)\*P4=0 in odd iterations; or
2. (P6vP7vP8)\*P8=0 in even iterations

Where "v" expresses the logic "OR" operation. C(P)=1 means P is 8-simple.

In other words, there is only one group of 8-connected 1's around P. Under this condition, deletion of P will not break the connectivity of the elements in the 3\*3 window under processing. Condition (a) guarantees P is not a break point. The Guo Hall algorithm is better in detecting the end points than the Zhang Suen algorithm. The use of N(P) allows one to identify the end points whether or not they have one or two 1's 8-neighbors[18].

### 3.2.6.3 Performance evaluation of thinning algorithm

**Thinning rate:** The aim of image thinning is to obtain the skeleton of the image. In order to evaluate the thinning algorithm, the thinning rate (TR) is :



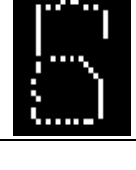

$$TR = \left(1 - \frac{T1}{T0}\right) * 100\%$$

Where T0 is the number of object pixel in original image, T1 is the number of object pixels in thinning image.

Generally, large TR indicates higher thinning degree [7]. A comparative analysis is presented in Table 3.

Since thinning rate obtained by Guo Hall algorithm is greater, it can be used for the thinning purposes.

Table 3: Comparative analysis of different thinning algorithms

Algorithm	Thinned Image	Thinning Rate (%)
Zhang Suen Algorithm		53
		28.7
Guo Hall algorithm		73
		56

## STEPS AFTER PRE-PROCESSING

After Pre-Processing is done, the underlying steps can further be added to the recognition process.

### 1. FEATURE EXTRACTION

Since many classifiers cannot process the raw images or data efficiently, extracting the relevant information of the data is needed. Feature extraction is the step that aims to extract relevant information by reducing the dimension of the data. Feature extraction stage in offline handwritten pattern recognition analyses these handwritten character segment and selects a set of features that can be used to uniquely identify

that character segment. The commonly used features are statistical, structural, moments and global transformations.

## 2. CLASSIFICATION

The classification stage is the decision making part of a recognition system and it uses the features extracted in the previous stage. The classification methods can be summarized in categories of statistical methods, artificial neural networks (ANNs), kernel methods, and multiple classifier combination.

## 4. CONCLUSION

The above defined pre-processing steps yield a furnished image that can be efficiently be used further for feature extraction and classification. The comparative study among different filters and thinning algorithms provides a good horizon towards the filtering and thinning steps. The above designed approach effectively pre-processes the image and therefore, makes the task of feature extraction and classification easier and simplified.

## 5. REFERENCES

- [1] Arica N., Yarman-Vural F.T. (2001),” An Overview of Character Recognition Focused on Off-Line Handwriting” (2001) IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 31, no. 2, pp.216-233.
- [2] Indira B., shalini M., Murthy Raman M.V., Shaik M.S. (2012) “Classification and Recognition of Printed Hindi Characters Using Artificial Neural Networks” I.J. Image, Graphics and Signal Processing, vol. 6, pp. 15-21.
- [3] Espana-Boquera S., Castro-Bleda M.J., Gorbe-Moya J., Zamora-Martinez F. (2011) “ Improving offline handwritten text recognition with hybrid HMM/ANN models” IEEE transactions on pattern analysis and machine intelligence vol.33, no.4, pp. 767-779.
- [4] Rajput K.Y., Mishra S. “Recognition and editing of devnagari handwriting using neural network” Proceedings of SPIT-IEEE Colloquium and International Conference, vol. 1, pp. 66-70.
- [5] Shrivastava S., Singh M.P. (2010) “ Performance evaluation of feed forward network with soft computing techniques for handwritten English alphabets” Applied soft computing ,Elsevier, vol.11 pp. 1156-1182.
- [6] Shenbagavadivu S., Dr. Devi M.R.(2013) “An investigation of noise removing techniques used in spatial domain image processing” International Journal of Computer Science and Mobile Computing Vol.2 Issue. 7, pp. 378-384.
- [7] Dancheng Xu, Bailiang Li, Nijholt A. (2009) “A novel approach based on PCNNs Template for fingerprint Image Thinning”, Eight IEEE/ACIS International Conference on computer and Information Science, pp. 115-119.
- [8] Goyal G., Dr. Dutta M., Er. Girdhar A. (2010) “A Parallel Thinning Algorithm for Numeral Pattern images in BMP Format.
- [9] R.Gonzalez and R.E. Woods, Digital Image Processing, Prentice Hall, 2011.
- [10] Rani R., Kaur K. (2013) “ Experiment analysis of different texture based features of image using simplified Gabor Gaussian Wavelet transform” International Journal of Engineering and Advanced Technology (IJEAT), Vol. 2, pp. 365-368.
- [11] K.purnima, T V Sampath Kumar, “Lossless Image Compression Using Traditional and Lifting Based Wavelet Transform” International Journal of Innovative Research and Studies. ISSN 2319 -9725.
- [12] Saeed K., Tabe Dzki M., Rybnik M., Adamski M. (2010) “K3M: A Universal algorithm for image skeletonization and a review of thinning techniques” International Journal of Applied Mathematics and Computer Science, Vol. 20, No. 2, pp. 317–335.
- [13] Xu D., Li. B., Nijholt A. (2009) “A novel Approach Based on PCNNs Template for Fingerprint Image Thinning” Eight IEEE/ACIS International Conference on Computer and Information Science, pp. 115-119.
- [14] Ahmed P. (1995) “A neural network based dedicated thinning method” Elsevier, pp. 585-590.
- [15] Shang L., Yi Z. (2007) “A class of binary images thinning using twp PCNNs” Elsevier, pp. 1096-1101.
- [16] Hallale S.B., Salunke G.D. (2013)” Offline and handwritten digit recognition using neural network” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 9.
- [17] Yadav D., Sanchez-Cuadrado S., Morato J. “ Optical character recognition for Hindi language using a neural network approach” (March 2013) J. Inf. Process Syst. vol. 9, no.1 pp 117-138.
- [18] Kumar H. and Kaur P. (2011)” A Comparative Study of Iterative Thinning Algorithms for BMP Images” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5), pp. 2375-2379.