

A Survey on Feature Extraction Methods for Handwritten Digits Recognition

Ishani Patel
Student (ME)
Department of CSE
LJIET, Ahmedabad

Virag Jagtap
Assistant Professor
Department of CSE
STBS, Surat

Ompriya Kale
Assistant Professor
Department of CSE
LJIET, Ahmedabad

ABSTRACT

Hand written digit recognition is highly nonlinear problem. Recognition of handwritten numerals plays an active role in day to day life now days. Office automation, e-governors and many other areas, reading printed or handwritten documents and convert them to digital media is very crucial and time consuming task. So the system should be designed in such a way that it should be capable of reading handwritten numerals and provide appropriate response as humans do. However, handwritten digits are varying from person to person because each one has their own style of writing, means the same digit or character/word written by different writer will be different even in different languages. This paper presents survey on handwritten digit recognition systems with recent techniques, with three well known classifiers namely MLP, SVM and k-NN used for classification. This paper presents comparative analysis that describes recent methods and helps to find future scope.

General Terms

Feature Extraction methods for digit recognition.

Keywords

Feature Extraction, Back Propagation (BP), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM).

1. INTRODUCTION

Due to the variations in style of writing the digits, it is sometimes difficult for the person to recognize digit. So if it is that much difficult for the person than think about how difficult it is when we replace human with a computer. So the recognition of handwritten digits by computer would be much difficult than recognition by human, because the computer do not have that much thinking power. If we want the machine to give accurate output, we need to make an accurate system. We need to provide a proper database from that machine can get train and then recognize accurately. We need to provide better method for feature extraction and better classifier for improving the accuracy of the system.

2. FUNDAMENTAL OF DIGIT RECOGNITION

Any recognition system's performance depend on two things: first it depend on how accurately the features are extracted from a digit image and second which classifier would be the best choice. The recognition system is applied in either offline or online mode. In offline recognition system we first generate the digit then we extract useful feature from the digit after that we store those features and then processing is done on it. While in the case of online recognition, digit is processed when someone created it. In this kind of recognition, external factor likes pressure speed of writing has very much influence on the system. Accuracy of Hand Written Digit Recognition is still limited due to large variation in shape, orientation, style, scale etc.

3. WORKING PRINCIPLE

Handwritten digit recognition system goes through following steps [14], Image acquisition, Preprocessing, Segmentation, Feature extraction, classification. Image acquisition means getting the image either by scanning documents or by capturing photograph or by directly writing using mouse.

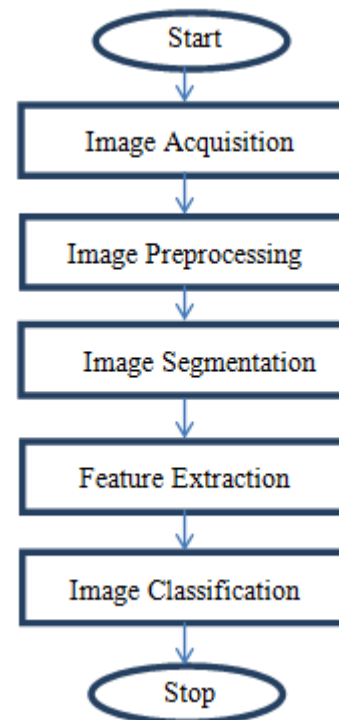


Fig 1: Flow chart for digit recognition system [14]

Image Preprocessing involves noise removal, converting image to black or white image, aligning document properly, and resizing image if too large. Image segmentation means to partition the image into multiple segments so to make it more meaningful and easy to analyze. Feature Extraction is said to be a heart of any pattern recognition system. Feature Extraction means to capture relevant characteristic of a target object. It is mainly related to dimension reduction. The extracted features from an image are applied as an input to the train classifier like neural network or support vector machine. Various Classifiers are now used for comparing the input with the stored pattern, hence to find best matching class.

4. FEATURE EXTRACTION TECHNIQUES

When the data as an input is too large and could not processed correctly at that time the data is transformed into reduced representation set of feature. The major goal behind such reduction is to improve recognition rate.

4.1 Diagonal based feature extraction [1]

In this Feature Extraction, the whole image is divided into equal zones, and then features are extracted from each zone by moving along diagonal of its irrespective pixels.

4.2 Hotspot Feature Extraction [4],[11]

In this method, the distance between black pixels and the hotspot in each direction is calculated.



Fig 2: Distance between hotspot and black/white pixels [4]

The feature value of the hotspot j is the coordinate of the hotspot, k is the distance from hotspot coordinate to the closest black pixel, i is the distance when no black pixel found in particular direction. In this technique, the size of the hotspot is defined as NxN. The distance between the black pixels and the hotspot is D_{si} and is calculated using following equation [4] given below:

$$D_{si} = \begin{cases} \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2} & \text{if } (x_i, y_i) \text{ exists,} \\ d_{max} & \text{else} \end{cases} \quad (1)$$

(x_s, y_s) is the coordinate of hotspot, (x_i, y_i) is the coordinate of black pixel. The direction of the hotspot is defined by the chain code directions [12] as shown in Fig 3. If there is no pixel found in the direction then distance is set to d_{max} .

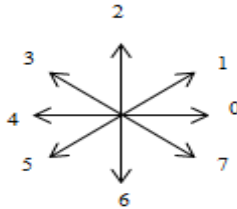


Fig 3: A 8 direction chain code [4]

4.3 Water Reservoir Method [2],[3]

This kind of feature extraction use water reservoir principle. If the water is dispensed from any side top, bottom, right, left, then the cavity regions of the component where the water will be stored are considered as reservoir [2]. The reservoir area is obtained when digit is not connected. Mainly four reservoirs are formed:

- Top reservoir: Reservoir obtained only when water is poured from top of the digit.
- Bottom reservoir: Reservoir obtained only when water is poured from bottom of the digit.
- Left reservoir: Reservoir obtained only when water is poured from left of the digit.
- Right reservoir: Reservoir obtained only when water is poured from right of the digit.

After getting the regions, the ratio of the reservoir of the pixels where water is stored is considered as total area and is computed in all four directions and hence stored as a feature vector. Fig 4 shows top reservoir area generated in the digits when unconnected from top.

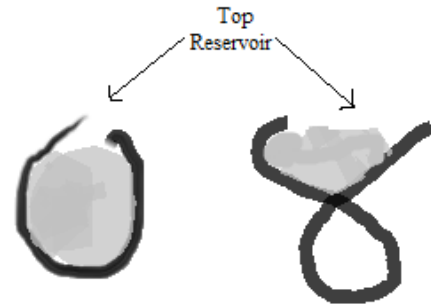


Fig 4: Top reservoir area [3]

4.4 Fill Hole Density [3]

A hole is a set of pixel touches the border such that it divides the background into two regions. To avoid this, it required to fill specific area to connectivity of background pixels. A 4-connected neighborhood connectivity rule [13] is used for connectivity of the background pixel. It says that the pixel that touches the center pixel and connected horizontally or vertically to the center pixel are said to be neighbor. Here 3x3 window is considered with set of 0's and 1's. The looping area of the digit is filled with ON pixels [13]. The ratio of the fill-hole density with total area estimate is taken as feature.

4.5 Contour Angular Technique [4]

This technique consists of two stages. In first stage the methods divide the handwritten character into 16-non overlapping blocks and the contour of the handwritten image is considered as 8-directional codes, see Fig. 3. Starting from any point S, the searching is done along the edges of the image blocks, first along the upper, the right, the bottom until finished at the left edge. For the image shown in Fig 5(a), the corresponding block image is shown by Fig 5(b).

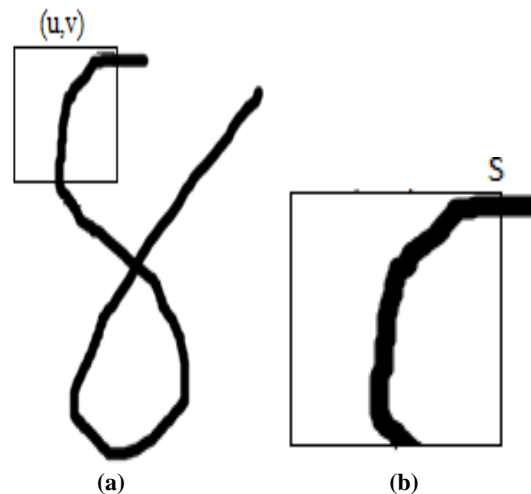


Fig 5: (a) Image block from whole image, (b) Starting from S point & searched along (u,v) border [4]

In the second stage, 8 directional codes (see Fig 3) are used for identifying the contour of neighbor pixels. The contour of the image is computed. This method considered two angles ψ_1 and ψ_2 as shown in Fig 6. Moving from S until the last pixel found in each block and then counts the co-occurring angles in a two- dimensional array indexed by ψ_1 and ψ_2 . Combining output from both stages will give a final feature vector.

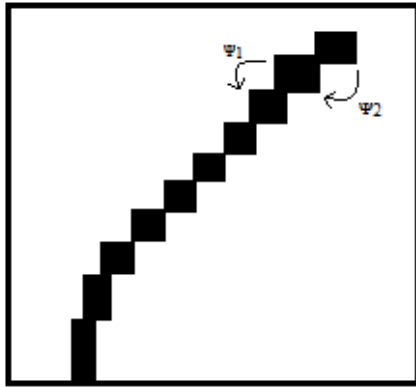


Fig 6: Angular occurrence of ψ_1 and ψ_2 [4]

4.6 Distance Profile [5]

In this type of feature extraction, a boundary box is fitted on each numeral. Image profile are then computed in all four direction- two in horizontal direction from left and right sides and in vertical direction from top and bottom side. Profiles in each direction are calculated by traversing of distance from the boundary box to the edges of character.

4.7 Image Centroid & Zone Centroid based [6]

In this method first compute the input image centroid then divide the image into n equal zones. Compute the distance between the image centroid to each pixel in the zone then average distance is computed. For n zone, n features are obtained. Now compute zone centroid of each zone. After computing the zone centroid the distance between zone centroid to each pixel in that particular zone is calculated. After computing distance for each zone centroid, average distance is computed. Finally, n features are obtained for the classification.

4.8 Hough Transform [7]

Hough transform method is used to detect lines, circles or other parametric curves. In this feature extraction method, first convert the input image into binary image of numerals. Divide the image into n equal zone. Calculate the value (x,y) of r for each ON pixel in each zone. The value of r is given as [7]:

$$r = x \cos \theta + y \sin \theta \quad (2)$$

For each ON pixel take the standard theta values 0,30,60,90,120,150. Then calculate the largest value of ON pixel in each zone then calculate the average of largest values. For each zone which doesn't have foreground pixel, consider feature values as zero for that zone.

4.9 Four View Projection Profile [7]

A Projection profile is a histogram of number of black pixel values those are accumulated along parallel lines taken through the image. In this type of feature extraction, first convert the gray scale image into binary image of numerals. Divide the image into n equal zones. For each zone find the horizontally, vertically and right diagonal projection profile in each zone. Store the average largest values means a peak value of each zones. These values are used as a feature vector.

5. CLASSIFIERS

After getting feature vector from the input image next step is to decide which classifier to use for classifying the class of

digits. The most traditional classifier to be used is neural network and other are SVM, k-NN.

5.1 Classification using Backpropagation Neural Network [9]

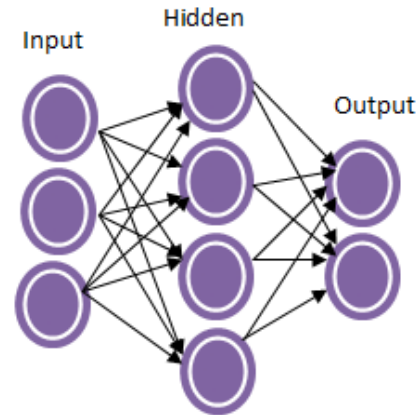


Fig 7: Neural Network [9]

In Handwritten digit recognition system, a Backpropagation neural network is used for training of the network. Using back propagation, network weights are updated accurately so as to minimize errors. While considering neural network as a classifier, three things need to keep in mind:

1. Number of Input neurons
2. Number of Hidden neurons
3. Number of Output neurons

Number of input neurons would be the number of reduced features extracted from the input image. Number of hidden neurons taken into hidden layer should be in the range between the size of input layer and the size of output layer or the number hidden neurons should be around 2/3 of input layer size, plus the size of output layer or the number of hidden neurons should be less than twice the input layer size [15]. Number of Output neurons would be the size of desired output of the system. For example, in case of recognition of hand written digits the output neurons would be 10 and in the case of recognition of handwritten alphabets the output neurons would be 26. For the design of a special class of layered feed forward networks known as multilayer perceptron (MLP), the back-propagation algorithm has emerged as the workhorse. The training of an MLP is usually accomplished by using a back-propagation (BP) algorithm that involves two phases:

- 1) Forward Phase: During this phase the parameters of the network are fixed. In this phase the input signal is propagated through the network of Fig. 7 layer by layer. The forward phase finishes with the computation of an error signal e,

$$e = DR - AO, \quad (3)$$

Where, DR is the desired response and AO is the actual output produced by the network in response to the any input x.

- 2) Backward Phase: During this phase, the error signal e which is generated in forward phase is propagated through the network of Fig. 7 in the backward direction, hence the name of the algorithm. During this phase the adjustments are applied to the

parameters of the network for minimizing the error e . The back-propagation learning algorithm is simple to implement and computationally efficient.

5.1.1 The Back Propagation [9]:

- Initialize the weights. Typically weights are initialized to small random number.
- For each training example in the testing set:

Input the training data to the input nodes, then calculate the output, which is the output of node k defined as O_k . This is done for each node in hidden layer (s) and output layer.

- Then calculate error Θ_k for each output node. The target of the node is defined as t_k :

$$\Theta_k \leftarrow O_k(1-O_k)(t_k-O_k) \quad (4)$$

- Now calculate for error at each hidden node is given by:

$$\Theta_k \leftarrow O_k(1-O_k) \sum_{k \in \text{child}(h)} W_{hk} * \Theta_k \quad (5)$$

W_{hk} is the weight from layer h to layer k .

- Finally adjust the weights of all the links, where X_i is the activation and ℓ is a learning rate:

$$W_{i,j} \leftarrow W_{i,j} + \ell \Theta_j X_i \quad (6)$$

The neural network needed to be trained at many iterations of the training set to find an acceptable approximation of the function on which it is going to be trained.

5.2 Classification using SVM [8]-[10]

A Support Vector Machine (SVM) is a classifier formally defined by a separating hyperplane. SVM is used for finding an optimal hyperplane which is then used for categorizes new examples for the given labeled training data. Only need to take into account in which sense the hyperplane obtained optimal. For a linearly separable set of 2D-points which belong to one of two classes, we need to find a straight line that separating them.

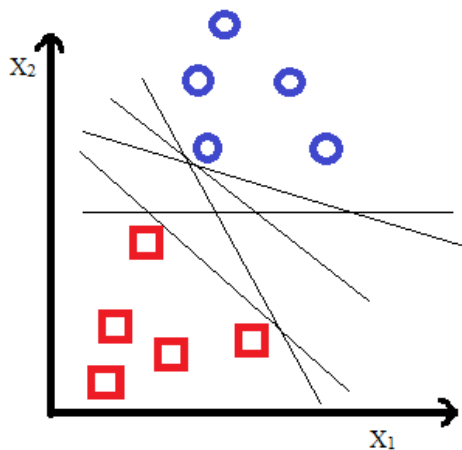


Fig 8: Straight lines that divide two classes [9]

In the above Fig 8 there exist multiple lines that offer a solution to the problem. A line is bad if it passes too close to the points because if it passes closely to the points then it will be noise sensitive and it will not generalize correctly. Therefore, the goal should be to find the line which is passing as far as possible from all points. The SVM algorithm is now

used for finding the hyperplane that gives the largest minimum distance to the training examples.

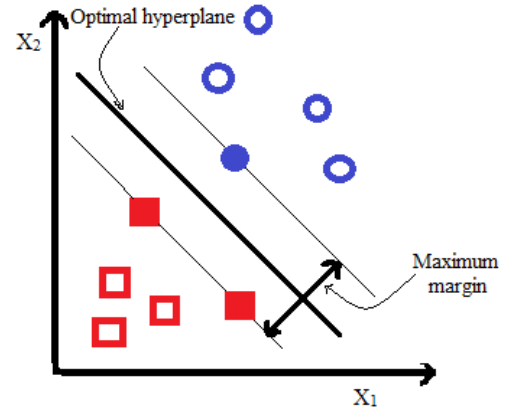


Fig 9: Maximum margin hyperplane for a SVM trained with samples from two classes [9]

Hyperplane be given as:

$$f(x) = B_0 + B^T X \quad (7)$$

where B =weight vector, B_0 = bias. The optimal hyperplane can be represented in infinite number of ways by scaling B_0, B . Suppose X is represented as training examples which close to hyperplane. Then the training example that are close to hyperplane are called support vector. This representation is known as canonical hyperplane. The distance between point X & hyperplane (B, B_0) is given by:

$$\text{distance} = |B_0 + B^T X| \quad (8)$$

For canonical hyperplane, the numerator is equal to one & distance to support vector is given by:

$$\text{distance}_{\text{support vector}} = \frac{1}{|B|} \quad (9)$$

M is the margin that is twice distance to the closest examples.

$$M = \frac{2}{|B|} \quad (10)$$

The problem of maximizing M is equivalent to the problem of minimizing a function $L(B)$ [16]. The constraint model the requirement for the hyperplane to classify correctly all the training examples X_i .

$$\min L(B) = \frac{1}{2} \|B\|^2 \quad (11)$$

$$\text{subject to, } y_i(B^T X_i + B_0) \geq 1, \quad \forall i \quad (12)$$

where y_i represents each of labels of training examples. This is a problem of Lagrangian Optimization [16] that can be used using Lagrange multipliers to obtain weight vector B and bias B_0 of the optimal hyperplane.

5.3 Classification using k-NN [9],[10]

The k-nearest neighbor algorithm is used for classifying the characters based on neighborhood in the feature space. The performance of KNN classifier is observed for different values of K ranges from 1 to 3. The Euclidean distances are computed between the test vectors to all the stored vectors.

This computed distances are then ranked in ascending order and from it the k nearest neighbors is chosen. The majority rule with the nearest point tiebreak is used for classification. A new case is classified by its K nearest neighbors which are measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

5.3.1 KNN Algorithm [9]:

Input: x(actual input), S(data set)

Output: class of x

for $(x', l') \in S$ do

{

1. Compute the distance $d(x', x)$.
2. Sort the $|S|$ distances by increasing order.
3. Count the number of occurrences of each class l_j among the k nearest neighbors.
4. Assign to x the most frequent class.

}

Table 1. Feature extraction methods with recognition rate and classifier

Sr . No.	Method	Classifier	Recognition Rate	Dataset	Year
1	Diagonal [1]	MLP	98.54	Own	2011
2	Hotspot [4],[11]	SVM	92.70	MNIST	2013
3	Water Reservoir [2],[3]	k-NN	96.94	MNIST	2014
4	Fill hole density [3]	k-NN	96.94	MNIST	2014
5	Contour Angular [4]	SVM	96.00	Own	2013
6	Distance Profile [5]	k-NN	94.58	Own	2012
		SVM	98.13		
7	Image Centroid & Zone Centroid [6]	MLP	94.60	Own	2008
		k-NN	94.70		
8	Hough Transform [7]	MLP	72.50	Own	2013
		SVM	93.12		
9	Four View Projection Profile [7]	MLP	98.73	Own	2013
		SVM	96.04		

Table 2. Feature extraction methods with training time, number of features & training samples

Sr. No .	Method	Training Time	Number of Features	Training Sample
1	Diagonal [1]	NM	69	1300
2	Hotspot [4],[11]	35ms	100	10000
3	Water Reservoir [2],[3]	NM	10	50000
4	Fill hole density [3]	NM	10	50000
5	Contour Angular [4]	37ms	192	10000
6	Distance Profile [5]	NM	128	1500
7	Image Centroid & Zone Centroid [6]	NM	100	1000
8	Hough Transform [7]	NM	25	1200
9	Four View Projection Profile [7]	NM	25	1200

*NM= Not Mention

Table 3. Prons & cons of different feature extraction methods

Sr. No.	Method	Merits	Demerits
1	Diagonal [1]	Gives more specific information	Not perform well if character builds by straight lines.
2	Hotspot [4],[11]	Novel approach with good accuracy	Additional overhead of calculating distance in all 8 directions if no pixel found.
3	Water Reservoir [2],[3]	Novel approach with good recognition rate	If the digit is not connected hence required additional feature extraction method (s).
4	Fill Hole Density [3]	Focus on neighborhood pixels only	Additional overhead, For good accuracy require additional extraction method.
5	Contour Angular [4]	Fast implementation of quantized angle co-occurrence computation.	If multiple neighbors then queue is used which makes it complex
6	Distance Profile [5]	Gives good information from each direction.	Training might slow for larger size image.
7	Image Centroid & Zone Centroid [6]	Provide good result even when preprocessing steps like filtering, smoothing & slant removal not performed, Merging zoning and centroid based method gives good accuracy. The goal of doing zoning is to focus on local characteristic rather than global.	Too many zones create chaos in the solution.
8	Hough Transform [7]	It gives robust detection under noise and partial occlusion.	Computationally complex for the objects with many parameters, Not give good accuracy.
9	Four View Projection Profile [7]	Good and accurate information gathered from all desired direction.	Range of values is large

6. CONCLUSION

This paper presents survey of recent feature extraction methods and classifiers. This survey shows four view projection profile method with MLP gives highest recognition rate. Diagonal based extraction with MLP and Distance Profile based extraction with SVM also gives good accuracy. Following issues with handwritten digit recognition system: 1) Most of the work has been done on smaller dataset. 2) Training time is not considered in most of the work. 3) Most of the work has been done for the English language because dataset is available like MNIST, but there isn't any efficient work has been done for the other languages with larger dataset. 4) Most of the recognition systems are implemented for single language. 5) Restriction to some feature extraction methods, they are local to language, not have global scope.

In future we can work on reducing the training time of the system. Here most of the work is done on CPU which performs serial sequential computing. That is why the time is not considered in most of the work because the training time is high. So for reducing the time in future, the system can be implemented on GPU with parallel processing.

7. ACKNOWLEDGMENTS

I am sincerely thankful to all who have contributed for this survey and providing me valuable & useful guidance on this work. I am thankful to my parents & my friends and all faculty members of LJJET (CSE Department) who push me to work hard and encourage me.

8. REFERENCES

- [1] J.Pradeep, E.Srinivasan And S.Himavathi,"*Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network*", IEEE Conference On Electronics Computer Technology, Feb 2011, Vol 3, No 1, Pages.27-38.
- [2] Water Reservoir Based Approach For Touching Numeral Segmentation ,U. Pal, A. Belaid,And C. Choisy , *Springer 2003*.
- [3] U Ravi Babu, Dr. Y Venkateswarlu, Aneel Kumar Chintha,"*Handwritten Digit Recognition Using K-Nearest Neighbor Classifier*" 2014 World Congress On Computing And Communication Technologies, Vol 4,Pages.60-65.
- [4] Olarik Surinta, Lambert Schomaker and Macro Wiering "A Comparison of Feature And Pixel- Based Methods For Recognizing Handwritten Bangla Digits " ,2013 12th International Conference On Document Analysis And Recognition, August 2013, Pages-165-169
- [5] Kartar Singh Siddharth, Renu Dhir, Rajneesh Rani, "Comparative Recognition of Hand Written Gurumukhi Numerals Using Different Feature Sets And Classifiers" , 2012 International Journal of Computer Applications, Pages. 20-24.
- [6] S.V Rajashearadhya, Dr P.Vanaja Ranjan "*Efficient Zone Based Feature Extraction Algorithm for Handwritten Numeral Recognition of Four Popular*

South Indian Scripts” Journal of Theoretical and Information Technology April, 2008, Pages.1171-1180.

- [7] Mamta Garg Et.Al. “A Novel Approach to Recognize the Off-line Handwritten Numerals using MLP and SVM Classifier”. IJCSET 2013, Vol. 4 No. 07 July 2013, Pages. 953-958.
- [8] S N Sivanandam, S Sumathi, S N Deepa “Introduction to Neural Networks Using MATLAB6.0” Mc Graw Hill Education 2013
- [9] Simon, Hykin “Neural Networks and Learning Machines”, PHP 2013
- [10] Han, Kamber, Pei “Data mining Concepts & Techniques”, Morgan Kaufman MIT press.
- [11] Olarik Surinta, Lambert schomaker and Macro Wiering , “Handwritten Character Classification Using the HotSpot Feature Extraction Techniques ” *ICPRAM 2012- International Conference on Pattern Recognition Application and Methods*, Pages. 261-264.
- [12] Imran Siddiqi, Nicole Vincent, “A set of Chain code based features for Writer Recognition” IEEE 2009 10th International Conference on document analysis and Recognition, Pages.981-985.
- [13] R.C Gonzal, R.E.Woods, “Digital Image Processing”, Pearson education, 2002
- [14] Manish Shah et al. “A literature review on handwritten character recognition” ISRJ, ISSN: 22307850 vol-3 March 2013
- [15] Wen Yu, N. Sanchez, “Advances in Computational Intelligence”
- [16] Springer Science & Business Media, 18-Aug-2009.O.L.Mangasarian, David R.Musicant, “Lagrangian Support Vector Machines”, Journal of Machine Learning research 1(2001) pages. 161-177.