

Digital Attendance Calculation System using Support Vector Machine & Image Processing

Rajib Bag, Gautam, Komal Das, P. Mohini Amma

Computer Science & Engineering
Dr. B.R. Ambedkar Institute of Technology
Port Blair, Andaman & Nicobar Island, INDIA

Abstract— This paper deals with recognizing and classifying handwritten characters and classifies it with respect to other characters in the method for attendance calculation system. Existing system of calculation of attendance contains manual calculation for the number of students present or absent in a class. In this paper we proposed a digitized method for calculating number of classes present and absent for a student which in the previous case is time consuming and may sometimes produce incorrect result in some cases. It uses features of a character extracted using feature extraction techniques. There are methods been proposed for feature extraction that can be simulated using support vector machine for further calculation of the attendance using the attendance register. Support vector machine is one of the techniques that are used to classify a handwritten character based on handwriting recognition and identify the digitized output Support vector machine classifies a character by using a large number of training set that needs to be fed into the database for the accurate output.

Key words—Support Vector Machine, Image Processing, Feature Extraction, Attendance Calculation System

INTRODUCTION

In the world, where everything has been digitized, using computer based algorithms and various coding and decoding techniques, data is processed with the help of programs and we get the desired output. Identification of objects in a real world plays a key role for human-computer interaction in a computer-augmented environment using augmented reality techniques. There are two characters used in this application. The main aim of this paper is to present an application using support vector machine that calculates the consolidated attendance of each student by the image of attendance register. This application has been developed using steps including pre-processing for image acquisition, feature extraction to extract the features from an acquired image and Support Vector Machine where these features are used to train a SVM classifier with different samples and a training set is developed. The extracted features is used to form a training set for Support Vector Machine, a test sample is checked to illustrate that whether a sample belongs to the class or not. Whenever a new character either P or A is given to the system it forms a testing set and classifies whether the character is P or A. The total number of P and A are calculated for each row.

PREPROCESSING

A. CONVERSION OF RGB TO BINARY

Binarization[8] is one of the most important techniques for preprocessing stage. Among many binarization techniques, the Otsu's method is considered as the most commonly-use done in the survey papers. If a pixel is greater than or equal to the threshold intensity, the resulting pixel is white ("0"). On the other hand, if a pixel in the image has intensity less than the threshold value, the resulting pixel is black ("1"). Binarization[8] of image consists of that either global or local threshold. Global thresholding has a good performance in the case that there is a good separation between the foreground and the background.

Otsu's global threshold method finds the global threshold t that minimizes the intra-class variance of the resulting black and white pixels. Then the binarization is formed by the setting

$$b_i = 1 \text{ if } x_i < t \text{ and } b_i = 0 \text{ if } x_i > t$$

Unlike global approaches, local area information may guide the threshold value for each pixel in local (adaptive) thresholding techniques. A local algorithm is introduced in that calculates a Pixel-wise threshold by shifting a rectangular window across the image.

The threshold T for the center pixel of the window is computed using the mean m and the variance s of the gray values in the window:

$$T = m + ks$$

Where k is a constant set to -0.2. The value of k is used to determine how much of the total print of object boundary is taken as a part of the given object.

B. MEDIAN FILTERS

The median m of a set of values is the value that possesses the property that half the values in the set are less than m and half are greater than m . Median filtering is the operation that replaces each pixel by the median of the grey level in the neighborhood of that pixel.

Median filters[8] are nonlinear filters because for two sequences $x(n)$ and $y(n)$

$$\text{median}\{x(n) + y(n)\} \neq \text{median}\{x(n)\} + \text{median}\{y(n)\}$$

Median filters[8] are useful for removing isolated lines or points (pixels) while preserving spatial resolutions. They perform very well on images containing binary (salt and pepper) noise but perform poorly when the noise is Gaussian.

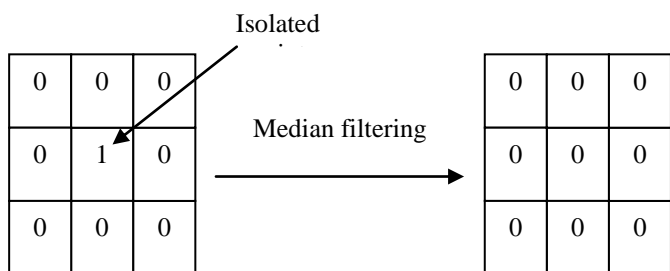


Fig 1 Median Filter

C. RESIZING

Resizing[8] an image is defined as reducing or enlarging dimensions of an image. The resizing is usually done described given below:

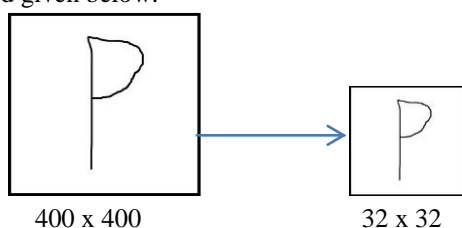


Fig 2 Image resizing

D. THINNING

Thinning [8] is an important preprocessing step in optical character recognition. The purpose of thinning is to delete redundant information and at the same time retain the characteristic features of the image. In order to reduce the quantity of information minimally, a thinning algorithm play an important role in recognition of the character, figure, and drawing.

Thinning [8] is usually involves removing points or layers of outline from a pattern until all the lines or curves are of unit width, or a single pixel wide. The result in set of lines or curves is called the skeleton of the object as shown in figure.

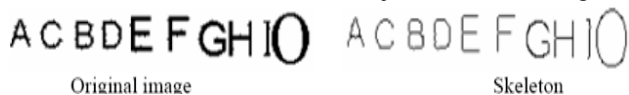


Figure 3: Skeleton produced by thinning process

EXISTING FEATURE EXTRACTION TECHNIQUES

A. FREEMAN CHAIN CODE

The main objective of feature extraction is to remove redundancy from data. The task of human expert is to select features that allow effective and efficient recognition of pattern. The architecture of recognition system

Feature extraction is a very important in recognition system because it is used by the classifier to classify the data. Chain code is one of the representation techniques that is useful for image processing, shape analysis and pattern recognition fields. Chain code representation gives the boundary of character image in which the codes represent the direction of where is the location of the next pixel. The first approach of chain code was introduced by Freeman in 1961 that is known as Freeman Chain Code (FCC)[16]. There are two directions of chain code, namely 4-neighborhood and 8-neighborhood.

B. CAEMASTRA TECHNIQUE

The present feature extraction[7] works on global and local features. The global feature includes the analysis of pattern with respect to baseline and width/height ratio. The local features divides the character by dividing into 4 x 4 grid and calculating the number of pixels horizontally and vertically along each column at regular pixels. A probability of number of pixels in the grid with respect to the number of pixels found by the same method with the whole image. If n_i is the number of foreground pixels in cell i and M is the total number of foreground pixels in the pattern, then the gray feature related to cell i is n_i/M . The other operators try to estimate to which extent the black pixels in the cell are aligned along some directions. For each direction of interest, a set of N , equally spaced, straight lines are defined, that span the whole cell and that are parallel to the chosen direction. Along each line $j \in [1, N]$ the number n_j of black pixels is computed and the sum $\sum_{j=1}^N n_j^2$ is then obtained for each direction. The first global feature measures the fraction of the character below the baseline and detects eventual descenders. The second feature is the width/height ratio. The number of local features can be arbitrarily determined by changing the number of cells or directions examined in each cell. Therefore the feature set was tested changing the number of cells and the grid giving the best results (4×4) was selected. A feature vector of 34 elements was formed. Two features are global (baseline and width/height ratio) while the remaining 32 are generated from 16 cells, placed on a regular 4×4 grid from each cell.

SUPPORT VECTOR MACHINE

Although there are several other algorithms[12][13][14] as pattern classifiers, Support Vector Machine (SVM) [2][3][4][7][16] is an algorithm that is widely used among them, after projecting data in the Feature space, computes an Optimal Separating Hyper plane, based on Statistical Learning Theory [5][13].

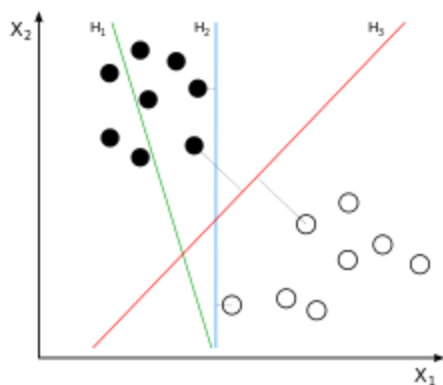


Fig 4. H2 provides the maximum margin hyperplane between two classes

A. LINEAR CLASSIFICATION USING SVM:

OPTIMAL SEPARATING HYPERPLANE:

We are given a set D of patterns $x_i \in R^n$ with $i = 1 \dots l$. each pattern x_i belongs to either of two classes $y_i \in \{-1, +1\}$. The patterns with output +1 are called positive patterns, while the others are called negative patterns. The goal is to establish the equation

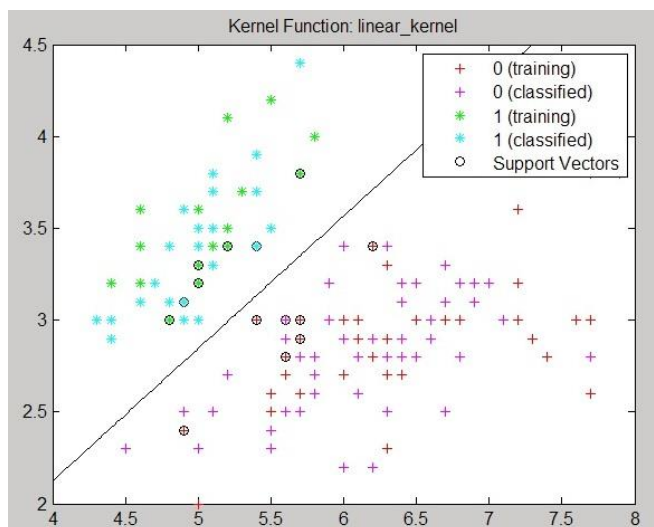


Fig 5. A linear SVM that separates two classes by a maximum margin hyperplane

$$w \cdot x + b \quad (w, x \in R^n, \quad b \in R)$$

of the Optimal Hyper plane that divides D leaving all points of the same class on the same side while maximizing the distance between the two classes and the hyper plane. This can be expressed by the constraints:

$$y_i(w \cdot x_i + b) > 1, \quad i = 1, \dots, l$$

In most practical problems a separating hyper plane may not exist. Due to presence of noise, an overlapping between classes may exist. Hence it has to allow that the constraints can be violated by some examples

Using slack variables $\epsilon_i > 0$ the constraints can be relaxed in:

$$y_i(w \cdot x_i + b) \geq 1 - \epsilon_i, \quad i = 1, \dots, l$$

Therefore we can construct a classifier, Support Vector Machine solving the following optimization problem

$$\begin{aligned} & \text{minimize} \\ & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \epsilon_i \\ & \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \epsilon_i, \quad i = 1, \dots, l \\ & \quad \quad \quad \epsilon_i > 0, \quad i = 1, \dots, l \end{aligned}$$

The classifier controls at the same time the margin $\|w\|$ and the number of training errors. The regularization constant $C > 0$ determines the trade-off between the two terms. The conditional optimization problem can be solved by introducing Lagrange multipliers $\epsilon_i > 0$ and a Lagrangian function

$$\min_{w, b} \max_{\alpha \geq 0} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(w \cdot x_i - b) - 1] \right\}$$

The Lagrangian L has to be minimized with respect to the primal variables w and b and maximized with respect to the dual variables i , i.e. a saddle point has to be found. The condition at the saddle point implies that the derivatives of L with respect to the primal variables must vanish

4.2. NON-LINEAR CLASSIFICATION USING SVM:

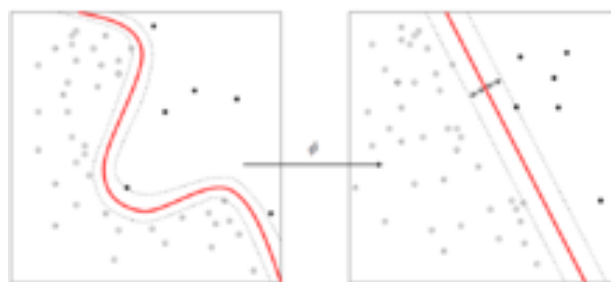


Fig 6. A nonlinear classification by using Kernel Functions

The original optimal hyper plane algorithm proposed by Vapnik in 1963 was a linear classifier. However, in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick (originally proposed by Aizerman et al.) to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function.

If the kernel used is a Gaussian radial basis function, the corresponding feature space is a Hilbert space of infinite dimensions. Maximum margin classifiers are well regularized, so the infinite dimensions do not spoil the results.

Some of the common kernel functions [6][10][15] include:

1. Polynomial(homogeneous): $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$
2. Polynomial(inhomogeneous):
 $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
3. Gaussian radial basis
function: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ for $\gamma > 0$.
Sometimes parameterized using $\gamma = 1/2\sigma^2$
4. Hyperbolic tangent: $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$,
for some $\kappa > 0$ and $c > 0$.

B. MULTICLASS SVM:

In order to use SVM when the number of classes K is larger than 2, a few different strategies have been suggested. In our experiments we have adopted *One-versus-rest (o-v-r)* method. The method learns one classifier for each of the K classes against all the other classes. More formally, the method consists in training K SVM classifiers f_j by labeling all training points having $y_i = j$ with +1 and $y_i \neq j$ with -1 during the training of the j th classifier. In the test stage, the final decision function $F(x)$ is given by:

$$F(x) = \text{argmax}(f_j(x))$$

PROPOSED METHODS

A. DISTANCE MEAN FEATURE EXTRACTION

There are various feature extraction techniques as given in [1][2][3][4][7]. The algorithm takes mean from all the sides by the distance from the first element pixel to the first black pixel in the row. The features are calculated as per the dimensions given below as four features gives four dimensions in an image. There are four features extracted from the character which are calculated as below:

Extracted distance from left to right

$$a = \frac{1}{n} \sum_{i=1}^n a_i$$

Extracted distance from left to right

$$b = \frac{1}{n} \sum_{i=1}^n b_i$$

Extracted distance from left to right

$$c = \frac{1}{n} \sum_{i=1}^n c_i$$

Extracted distance from left to right

$$d = \frac{1}{n} \sum_{i=1}^n d_i$$

These four features extracted are given as parameters to Support Vector Machine for further simulation.

B. FUDICIAL POINTS FEATURE EXTRACTION

The yet another technique for feature extraction is quiet simplified as it works for only P and A as the features is as per two characters. The features of the character are taken as per below:

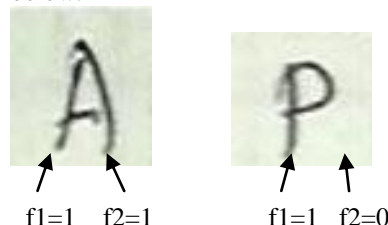


Fig 7. Fudicial Point Feature Extraction

1. The algorithm works as follows:

1. Start scanning from left bottom corner of the image.
2. If a black pixel $A[i][j]=1$ is found along the row then iterate $i=i+1$ and scan next row until $A[i][j]=1$ is found or $f1=1$.
3. If a black pixel $A[i][j]=1$ is found along horizontal scanning assign $f1=1$ and search for $f2=1$ in the same row.
 - a. If $f2=1$ in the same row then assign features as 1 and 1.
 - b. If $f2=0$ along the same row, then scan for the next 5 rows with iterations and search for $f2=1$ as given in the above figure, if not found assign $f2=0$.
4. Note the features $f1$ and $f2$ as parameter to Support Vector Machine.

2. Pseudocode

1. Algorithm FeatureExtraction(I2)
2. {
3. $s:=\text{size}(I2)$;
4. $f1:=0$;
5. $f2:=0$;
6. $\text{row}:=s(1)$;
7. while($\text{row}>0$)
8. {
9. for $\text{col}=1$ to $s[2]$
10. {
11. $x:=I2[\text{row},\text{col}]$;
12. if($x==0$)
13. {
14. $f1:=1$;
15. break;
16. }
17. }
18. if($f1==1$)
19. {
20. break;
21. }

```

22. row:=row-1;
23. }
24. for c=col+10: to s[2]
25. {
26. y:=I2[row,c];
27. if(y==0)
28. {
29. f2:=1;
30. break;
31. }
32. if(f2==1)
33. {
34. break;
35. }
36. }
37. if(f2~=1)
38. {
39. r:=row-1;
40. while(r>=row-3)
41. {
42. for c=col+10 to s(2)
43. {
44. y1:=I2[r,c];
45. if(y1==0)
46. {
47. f2:=1;
48. break;
49. }
50. }
51. if(f2==1)
52. {
53. break;
54. }
55. r:=r-1;
56. }
57. }
58. if(f2~=1)
59. {
60. r:=row-1;
61. while(r>=row-3)
62. {
63. c:=col-10;
64. while(c>0)
65. {
66. y1:=I2[r,c];
67. if(y1==0)
68. {
69. f2:=1;
70. break;
71. }
72. c:=c-1;
73. }
74. if(f2==1)
75. {
76. break;
77. }

```

```

78. r:=r-1;
79. }
80. }
81. A:=[f1,f2];
82. write f1,f2;
83. }

```

V. ATTENDANCE CALCULATION SYSTEM

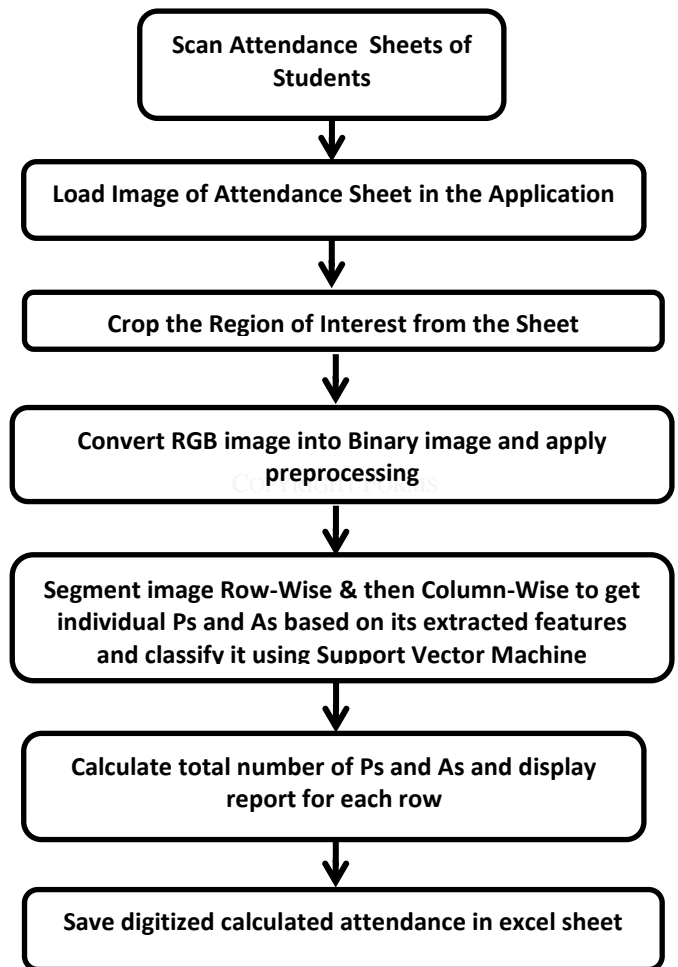


Fig 8: Flowchart for application for attendance calculation system

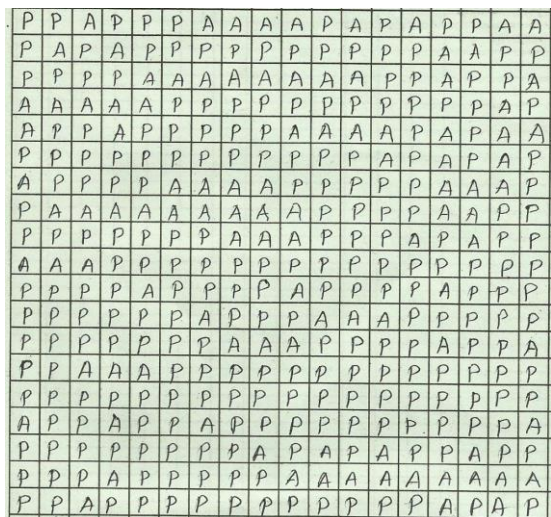


Fig 9. Sample of a cropped region from an Attendance sheet

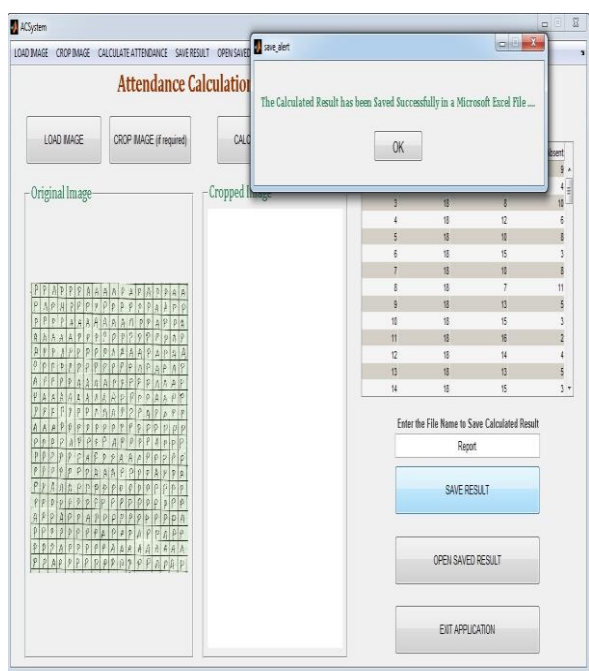


Fig 10. Snapshot of the application

VI. TEST CASE AND RESULTS

A. PERFORMANCE EVALUATION

Sl. No	Samples taken(120)	Image fed for testing	Number of images classified
1.	Sample 1	30	30
2.	Sample 2	30	30
3.	Sample 3	30	29
4.	Sample 4	30	29

Table 1. Performance Evaluation (Fudicial Point Method)

Sl. No	Samples taken(120)	Image fed for testing	Number of images classified
1.	Sample 1	30	30
2.	Sample 2	30	30
3.	Sample 3	30	28
4.	Sample 4	30	28

Table 2. Performance Evaluation (Distance Mean Method)

VII. CORRECTNESS VERIFICATION

The application is developed in MATLAB. The following are the accuracy results for the two methods in the analysis of the two proposed methods of feature extraction:

Sl. No	Name of the Proposed Method	No. of training Image	No of testing image	Accuracy rate
1.	Fudicial Point Feature Extraction	128	127	99.21%
2.	Distance Mean Feature Extraction	128	125	97.65%

Table 3. Correctness Verification

IX. CONCLUSION AND FUTURE WORK

This system proposes an algorithm that can achieve a task of scanning an attendance sheet that will speed up the process in the current system. This makes a manual work lot easier as user just need to scan an attendance register and calculation of attendance for each student will be done automatically. The simple user interface allows any user to easily operate the work. The user could also save the results for further work. The user could work easily using the system as this system is useful for the case where there are a large number of data available for calculation. The calculation time for support vector machine to identify is not too large as it uses only two characters for the classification of a character. There are a large number of data set available as a training set for accurate recognition of a character in support vector machine. This proposed system could reduce the work load of calculation of attendance from the attendance register.

There may be a mobile application which will capture attendance register and will produce the same result. This will be more easy of the calculation for a faculty. Moreover similar system can be developed for other languages as well.

ACKNOWLEDGMENTS

First of all we would like to thank the Almighty, who has always guided us to work on the right path of the life. Our greatest thanks are to our parents who bestowed ability and strength in our, to complete this work. This work would not have been possible without the encouragement of principal, Dr.B.R.Ambedkar Institute of

technology, PortBlair. His enthusiasm and optimism made this experience both rewarding and enjoyable.
We are equally grateful to HOD,C.S.E, Dr.B.R.Ambedkar Institute of technology, PortBlair who always encouraged us to keep going with work and always advised with his invaluable suggestions.

We would like to express our sincere gratitude towards, entire faculty and staff members of Dr. B.R.Ambedkar Institute of Technology, PortBlair for their direct-indirect help, cooperation, love and affection for completion this work.

REFERENCES

- [1]. R. Plamondon, S. Srihari, On-line and off-line handwriting recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. (2000) 63–84.
- [2]. A. Senior, A. Robinson, An off-line cursive handwriting recognition system, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 309–321.
- [3]. G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real time applications, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 366–379.
- [4]. R. Bozinovic, S. Srihari, Off-line cursive script word recognition, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1) (1989) 69–83.
- [5]. V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
- [6]. J. Shawe-Taylor, N. Cristianini, Kernels Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.
- [7]. F. Camastra, “A SVM-based cursive character recognizer,” in Pattern Recognition, vol. 40, pp. 3721-3727, March 2007.
- [8]. J. R. Parker, Algorithms for Image Processing and Computer Vision, John Wiley and Sons, 1997.
- [9]. Mathworks Matlab™ – The language of technical computing. Link: <http://www.mathworks.com/products/matlab/>
- [10]. SINGIRESU S. RAO, “Engineering Optimization”, A Wiley -Interscience Publication John Wiley & Sons, Inc.
- [11]. Cleber Zanchettin, Byron Leite Dantas Bezerra and Washington W. Azevedo, “A KNN-SVM Hybrid Model for Cursive Handwriting Recognition”, WCCI 2012 IEEE World Congress on Computational Intelligence June, 10-15, 2012 - Brisbane, Australia.
- [12]. Duda, Hart, and Stork, "Pattern classification" Chapter 5, Wiley, 2000.
- [13]. Vapnik V. N. “An overview of statistical learning theory”, IEEE Trans. on Neural Networks, 10(5):988–999, 1999.
- [14]. Haykin, S. “Neural networks - a comprehensive foundation”, Prentice Hall, 1996.
- [15]. J. Smola and B. Schölkopf. Learning with Kernels. The MIT Press, Cambridge, MA, 2002.
- [16]] R.Aruna devi, S.Latha, K.Vidhya, ”Handwritten Character Recognition Using Based Segmentation And Support Vector Machine Classifier” IJART, Vol. 2 Issue 1, 2012.”

Correspondence Author – Rajib Bag,
bag.rajib@gmail.com .

AUTHORS

First Author – Rajib Bag, M.Tech, DBRAIT and
bag.rajib@gmail.com

Second Author – Gautam, qualifications, DBRAIT and
gs0811@gmail.com.

Third Author – Komal Das, B.Tech, DBRAIT and
komal.das12990@gmail.com

Fourth Author – P. Mohini Amma, B.Tech, DBRAIT and
mohiniexpo@gmail.com