International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0047
ISSN (Online): 2279-0055

International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Classification of Handwritten Ancient Tamil characters using Complex Extreme Learning Machine

N.Sridevi [1], Dr.P.Subashini[2]
[1]Research Scholar, [2]Professor
Department of Computer Science
Avinashilingam University for Women, Coimbatore, Tamil Nadu, India.

Abstract: *Classification is the problem of identifying, to which set of categories a new observation belongs, on the basis of a training set whose category membership is known. The process of handwritten script classification involves extraction of some defined characteristics called features to classify an unknown handwritten character into one of the known classes. Zernike moments and regional features are extracted from the Tamil characters and they are formed as feature vectors. Complex Extreme Learning Machine is used to classify the handwritten ancient Tamil characters. Complex Extreme Learning Machine is trained with feature vectors. From the experimental result it is observed that the classifier when trained by combining Zernike moments with regional features gives a highest classification accuracy of 82.63%.*

***Keywords**: Zernike moments, Regional features, Complex Extreme Learning Machine, Classification, Handwritten Characters.*

I. Introduction

Tamil is one of the oldest languages know in India dating back over to two millennia. Inscriptions were written in olden days which are rich in knowledge related to different fields like literature, astrology, medicine, history and so on. Getting information from these inscriptions is essential. To do this, the common man must know to read the ancient Tamil scripts, whose writing styles and variations are different. This lays the main reason to carry out this research work in classification of ancient Tamil scripts. Classification is the problem of identifying to which set of categories a particular character belong on the basis of training set. The process of classification involves extraction of some definable information called features to classify an unknown handwritten character into one of the known classes.

From the literature, it has been observed that the pervious researchers have used traditional algorithms like SVM, SOM, HMM and two layer feed forward networks [1– 3]. These algorithms are far slower because the parameters must be tuned iteratively which increase the time taken for classification. Hence, to overcome this drawback of the traditional algorithms, this research work intends to use Complex Extreme Learning Machine (CELM) for classification of handwritten ancient Tamil scripts because in CELM the parameters are generated randomly and this saves the manually tuning of the parameters. The organization of the paper is as follows: In section 2, features extracted from the characters are explained in brief. Classification using CELM is given briefly in section 3. Results and discussion are discussed in section 4 and finally section 5 concludes this paper.

II. Feature Extraction

Feature extraction is defined as the process of extraction information from the raw data which is useful for classifying the unknown type into known class. Features are classified into two groups, they are structural features like strokes, end points, etc., and statistical features which are derived from the statistical distribution of points like zoning, moments, etc., [4]. Here statistical feature (Zernike moments) along with regional features are taken for classification of handwritten ancient Tamil scripts. In feature extraction each character is represented as a feature vector, which becomes its identity [5]. Therefore, in this paper features are extracted and formed as different feature vectors.

A. Zernike Moments

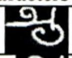
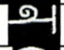
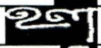
Zernike moments are a class of orthogonal moments. They are rotation invariant. The Zernike polynomials are a set of complex orthogonal polynomials defined over the interior of a unit circle [6]. Zernike moments are the projections of the image function onto these orthogonal basis functions. The Zernike moment of order n with repetition m for an image is given by

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y B(x,y)[v_{nm}(\rho,\theta)]^* \quad (1)$$

where, * is the complex conjugate operator

Table.1 shows the features obtained by using Zernike moments for the sample ancient handwritten Tamil scripts

Table 1: Zernike Moments of order 6

Characters	Moment 0	Moment 1	Moment 2	Moment 3	Moment 4	Moment 5	Moment 6
	3.97	2.72	19.65	3.33	17.53	5.49	-0.54
	3.51	1.10	6.67	0.38	-8.11	-0.40	-12.03
	3.53	0.65	1.91	5.58	0.81	4.05	3.48
	4.60	1.53	86.56	28.39	-26.19	34.52	-43.34
	3.59	0.02	7.31	1.12	23.14	-0.06	29.79
	3.73	6.43	2.04	2.88	3.58	4.88	-4.72
	5.05	7.22	6.68	0.16	5.07	0.43	-3.27
	5.18	7.68	5.31	1.62	-1.38	3.60	4.34
	3.00	0.78	1.89	0.24	1.16	0.19	2.87

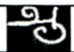
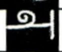
B. ***Regional features***

Here the features like Centroid, Orientation, Eccentricity, Extent, Mean and Standard deviation of Tamil characters are calculated

- Centroid – Specifies the center of mass of the region.
- Eccentricity – the eccentricity of the ellipse that has the same second-moments as the region.
- Orientation – This is the measure of angle in degrees between the x-axis and the major axis.
- Extent – It is defined as the ratio of pixels in the region to pixels in the total bounding box.
- Mean – Mean value of a region.
- Standard Deviation – It is defined as the measure of the dispersion of a set of data from its mean.

Table 2 shows the regional features extracted from the sample ancient handwritten Tamil scripts

Table 2: Regional Features

Characters	Centroid	Eccentricity	Orientation	Extent	Mean	Standard deviation
	828.00	43.89	23.73	0.82	0.22	472.09
	655.00	40.52	19.35	0.90	0.20	365.23
	700.00	28.05	23.37	0.47	0.29	409.30
	664.00	22.00	23.86	26.34	5.40	0.70
	1758.00	3.00	1.00	1.00	18.55	8.00
	471.00	16.49	27.02	0.89	0.29	272.65
	671.00	19.50	25.86	0.73	0.39	248.55
	524.00	22.68	18.28	0.59	0.22	293.58
	346.00	39.00	16.10	20.22	3.64	0.84

III. Classification

Tamil is one of the oldest languages known in India. Tamil has 247 characters in it out of which 12 are vowels, 18 are consonants, and 216 are composite characters (formed by combination of vowels and consonants) and one special letter known as Aydham. Tamil scripts are classified according to these groups. Target of the classification is to reduce the number of possible characters for an unknown character, from the known one [7]. Here, complex extreme learning machine is used for classification of handwritten ancient Tamil scripts.

A. *Complex Extreme Learning Machine*

Given a series of training samples (z_i, y_i), where i = 1, 2 ... N, $z_i \in C^n$ *and* $y_i \in C^m$, the outputs of the single hidden layer feed forward network with complex activation function for these N training data is given by[8]

$$\sum_{k=1}^{\bar{N}} \beta_k g_c(w_k.z_i + b_k) = o_i, \quad i = 1,\ldots,N \tag{2}$$

Where, $w_k \in C^n$ is the complex input weight vector connecting the input layer neurons to the hidden neuron, $\beta_k = [\beta_{k1}, \beta_{k2}, \ldots \beta_{km}]^T \in C^m$ is the complex output weight vector connecting the hidden neuron and the output neurons and $b_k \in C$ is the complex bias of the k[th] hidden neuron. g_c is a fully complex activation function. The above N equations can be written as

$$H\beta = O \tag{3}$$

and the number of hidden neurons is usually less than the number N of training samples.

$$H(w_1,\ldots,w_{\bar{N}}, z_1,\ldots,z_{\bar{N}}, b_i,\ldots b_{\bar{N}}) \tag{4}$$

$$= \begin{bmatrix} g_c(w_1.z_1 + b_1) & \cdots & g_c(w_{\bar{N}}.z_1 + b_{\bar{N}}) \\ \vdots & \cdots & \vdots \\ g_c(w_1.z_N + b_1) & \cdots & g_c(w_{\bar{N}}.z_N + b_{\bar{N}}) \end{bmatrix}_{N \, x \, \bar{N}} \tag{5}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\bar{N}}^T \end{bmatrix}_{\bar{N} x m}, O = \begin{bmatrix} o_1 \\ \vdots \\ o_N^T \end{bmatrix}_{N x m} \; and \; Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N x m} \tag{6}$$

Here, the complex matrix H is called the hidden layer output matrix. For fixed input weights and hidden layer biases, least squares solution of the linear system with minimum norm of output weight can be obtained [9]. The resulting least square solution is given by

$$\hat{\beta} = H^{\dagger} Y \tag{7}$$

where, $H^{\dagger}$ is the Moore-Penrose generalized inverse of complex matrix H.

B. *Activation Function Type*

The complex activation functions can be used in complex extreme learning machine. These include circular functions, inverse circular functions, hyperbolic functions and inverse hyperbolic functions [10]. In this research work, "arcsinh" is used as the activation function.

$$\arcsin h(z) = \int_o^z dt / ((1+t^2)^{1/2}), where\ z \in C \qquad (8)$$

C. *Moore-Penrose Generalized Inverse Matrix*

Matrix A is the Moore Penrose generalized inverse of complex matrix B, if ABA = A, BAB = B, (AB)* = AB, (BA)*=BA

Singular value decomposition (SVD) is used to calculate the Moore-Penrose generalized inverse of H. SVD is a factorization of a real or complex matrix which is used in many signal processing and statistics applications. The singular value decomposition of an mxn complex matrix M is a factorization of the form

$$M = U \Sigma V^* \qquad (9)$$

where, U is a mxm complex matrix, Σ is an mxn diagonal matrix with non negative real numbers on the diagonal and V^* is an nxn complex matrix

D. *The general of algorithm for CELM*

Given a training set N, complex activation function $g_c(z)$ and hidden neuron number $\tilde{N}$

Step 1: Randomly choose the complex input weight w_k and the complex bias b_k, where k = 1,...., $\tilde{N}$.
Step 2: Calculate the complex hidden layer output matrix H
Step 3: Calculate the complex output weight β using eq 7

The following procedure describes how the complex extreme learning machine is used in classification of handwritten ancient Tamil scripts. Training data and testing data are given as input the CELM.

Step 1: From training and testing data sets, the class labels are extracted and it is saved as Target vector.
Step 2: Complex random numbers are generated for the input weight of size (number of Hidden neurons X number of input neurons).
Step 3: Bias of hidden neurons are randomly generated from the complex numbers.
Step 4: Hidden layer output matrix H is calculated using the "arcsinh" activation function for the training data.
Step 5: Moore Penrose inverse matrix $H^\dagger$ is calculated
Step 6: Output weight is calculated using Eq 7.
Step 7: To find the actual output of the training data, the output weight is multiplied with $H^\dagger$.
Step 8: Repeat steps 5 to 8 to calculate the output of testing input.
Step 9: Classification accuracy for training and testing data is calculated.

IV. Results and Discussion

To classify the handwritten ancient Tamil characters, a sample of 500 characters is taken and their features are extracted. In order to train the Complex extreme learning machine, three different feature vectors are formed from the Zernike moments (FV1), Regional features (FV2) and by combining Zernike moments with regional features (FV3). Out of these 500 characters, 300 characters are used for training the complex extreme learning machine and remaining 200 characters are used as testing data. Number of hidden neurons is fixed as 50 neurons. The classification accuracy obtained by complex extreme learning machine using different feature vectors is tabulated.

Table 3: Classification accuracy obtained by using different feature vectors

Feature Vector	Training Time	Testing Accuracy
FV1	0.0313	78.84
FV2	0.0123	80.79
FV3	0.0569	82.63

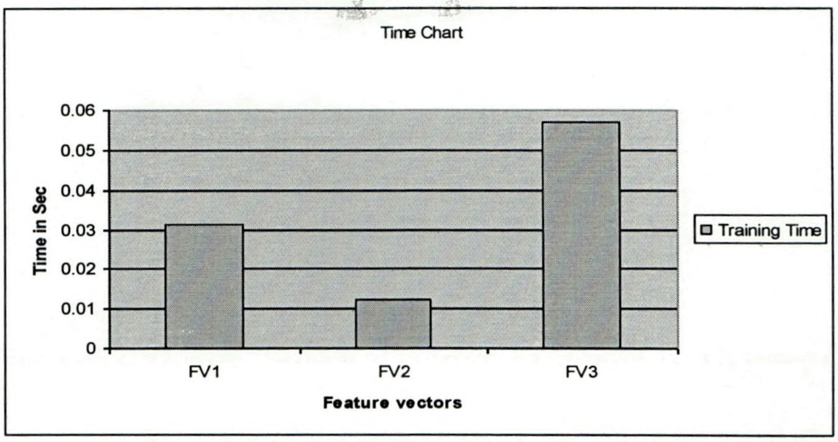


Figure 2: Time taken by CELM when trained using different vectors

Fig 2 shows that training time taken by CELM when trained with different feature vectors. It is observed that, for training the vector FV2 the CELM takes less time when compared to other feature vectors. The training time taken for FV3 is higher than other two vectors because number of attributes in this vector is more (i.e. 12). From the figure 3, it is found that when CELM trained using FV3, gives highest accuracy rate of 82.63 compared to others even though the time taken for training is more.

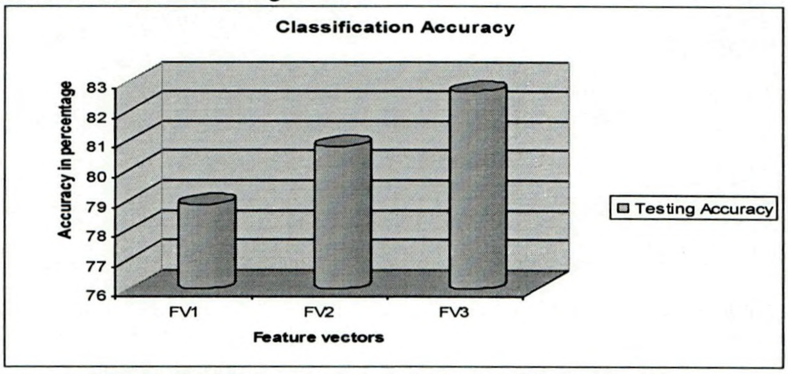


Figure 3: Classification accuracy obtained using various feature vectors

V. Conclusion

Complex extreme learning machine is used for classification of handwritten ancient Tamil scripts. Zernike moments and regional features are extracted from the Tamil characters and they are formed as feature vectors. These feature vectors are trained using CELM. It is found from the experimental results, that when Zernike moments combined with regional features and used as a feature vector for training, a highest classification accuracy rate of 82.63% is obtained. The main advantage of using CELM is that, the parameters need not be tuned manually and iteratively and more over the solution obtained is unique least norm square solution. Different combinations of CELM can be used which helps to increase the classification accuracy further.

References

[1] Venkatesh J and Sureshkumar C, " Tamil Handwritten Character Recognition using Kohonon's Self Organizing Map", International Journal of Computer Science and Network Security, Vol.9,No.12,pp.no:156-161,2009.

[2] Jagadeesh Kannan R et al,"Off-Line Cursive Handwritten Tamil Character Recognition", International Conference on Security Technology, IEEE Computer Society,pp.no:159-164,2008.

[3] Rajakumar S and Subbiah Bharathi V, " 7th Century Ancient Tamil Character Recognition from Stone Inscriptions", Indian Journal of Computer Science and Engineering, Vol.3,No.5,pp.no:673-677,2012.

[4] Heutte L et al, "A structural/statistical features based vector for handwritten character recognition", Pattern Recognition Letters 19, pp.no: 629 – 641, 1998.

[5] Sridevi .N and Subashini .P, "Moment Based Feature Extraction for Classification of Handwritten Ancient Tamil Scripts", International Journal of Emerging trends in Engineering and Development, Vol.7, No.2, pp.no: 106-115, 2012.

[6] Teague .M.R, "Image Analysis via the General Theory of Moments", Journal of the Optical Society of America, Vol.70, No.8, pp.no:920-930, 1979.

[7] Dr.N.Sengottaiyan et al, " Handwritten South Indian Language Recognition using Artificial Neural Network", International Journal of Advanced Research in Technology, Vol. 1, No.1,pp.no: 87 – 90, September 2011.

[8] Ming-Bin Li et al, "Fully complex extreme learning machine", Neurocomputing 68, pp.no: 306 – 314, 2005.

[9] Guang-Bin Huang et al," Extreme learning machine: Theory and applications", Neurocomputing 70, pp.no: 489 – 501, 2006.

[10] Kim .T and Adali .T,"Approximation by fully complex multilayer perceptrons", Neural Computing 15 (7), pp.no: 1641 – 1666, 2003.

International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-4, September 2011

An Optimal Binarization Algorithm Based on Particle Swarm Optimization

P.Subashini, N.Sridevi

Abstract— **Document binarization is an active research area for many years. Binarization algorithms play an important role in the preprocessing phase of any character recognition system. This paper compares several alternative binarization algorithms for handwritten documents, by evaluating their performance. The algorithms evaluated are, global thresholding, Otsu thresholding, Kittler-Illingworth and local thresholding, Niblack algorithm along with the proposed PSO algorithm. From the tests and results, we can wrap up with the assumption that the proposed algorithm shows improved results.**

Index Terms—**Evaluation, Global thresholding, Image Binarization, Local thresholding, PSO.**

I. INTRODUCTION

Binarization or thresholding is the process that converts an image into black-and-white: a threshold value is defined and the colors above that value are converted into white, while the colors below it are converted into black. This is a very simple process in digital image processing when one has a document with black ink written on a white paper. Document image binarization is an important step in the document image analysis and recognition pipeline. The performance of a binarization technique directly affects the recognition analysis [7]. The quality of the images however has a significant impact on the OCR performance, since most historical archive document images are of poor quality due to aging and discolored cards and ink fading. The PSO, first introduced by Kennedy and Eberhart is a flexible, robust, population based stochastic search/optimization algorithm with inherent parallelism [3, 11]. In recent years this method has gained popularity over its competitors due to its simplicity, superior convergence characteristics and high solution quality. This paper presents Otsu thresholding based on particle swarm optimization (PSO) algorithm. The binarization methods that are used in the evaluation are described in Section 2 and Section 3 describes the tests that were run and their results. The paper concludes in Section 4.

Manuscript received July 09, 2011.

Dr.P.Subashini, Department of Computer Science, Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatorte, India, 9442271971., (e-mail: mail.p.subashini@gamil.com).

N.Sridevi, Department of Computer Science, Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatorte, India, 9043126984., (e-mail: sridevi.n78@gnaul.com).

II. BINARIZATION ALGORITHMS

In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray scale or color images as binary images by picking a threshold value. Binarization algorithms are classified into global and local methods [10]. Fig 1 shows the block diagram of the binarization method.

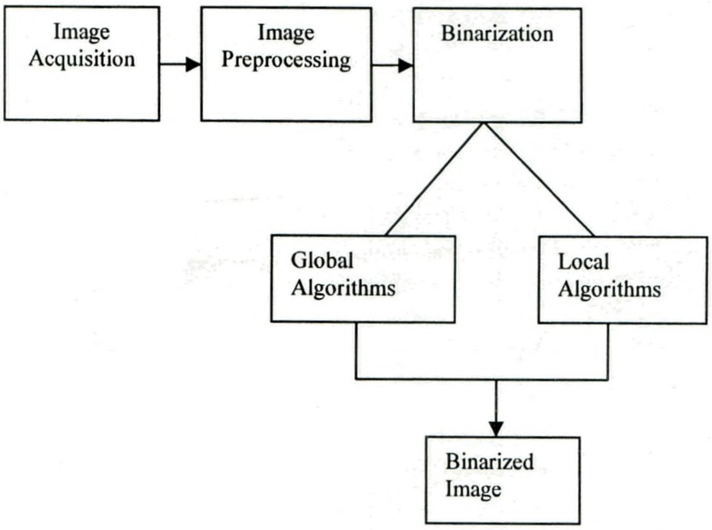


Figure 1: Block Diagram of Image Binarization Method

The global algorithms calculate one threshold for the entire image. The pixels are separated into two classes, foreground and background [1]. This can be expressed as in the equation (1).

$$I_b(x,y) = \begin{cases} black & if \quad I_f(x,y) \leq Thr \\ white & if \quad I_f(x,y) > Thr \end{cases} \tag{1}$$

where, $I_f(x,y)$ is the pixel of the input image and $I_b(x,y)$ is the pixel of the binarized image. While the local thresholding algorithms calculate different threshold values depending on the local regions of the image. A threshold value can be derived for each pixel in the image, and the image can be separated into foreground and background [1] which can be expressed as given in equation (2).

$$I_b(x,y)=\begin{cases} black & if \quad I_f(x,y)\le Thr(x,y) \\ white & if \quad I_f(x,y)>Thr(x,y) \end{cases} \qquad (2)$$

A. Global Thresholding Methods

a. Otsu Thresholding

Otsu is an often used global thresholding method. It is based on treating the gray level intensities present in the image as values to be clustered into two sets, one foreground (black) and one background (white) [9]. To carry out this, the algorithm minimizes the weighted sum of within-class variances of the foreground and background pixels to establish an optimum threshold. This is equivalent to maximizing the between-class scatter. From this a scalar number, K, is returned. This is then used to binarize the image through the following equation (3)

$$I_{bin}(x,y)=\begin{cases} 1, & if \ I_{gray}(x,y)\le K \\ 0, & if \ I_{gray}(x,y)\le K \end{cases} \qquad (3)$$

b. Kittler and Illingworth

Kittler and Illingworth present an algorithm that is based on the fitting of the mixture of Gaussian distributions and it transforms the binarization problem to a minimum-error Gaussian density fitting problem [6]. Assume that t is a threshold value used to segment the image into background and foreground, both of which are also modeled by Gaussian distributions, $p_B(t)$ and $p_F(t)$, respectively. Define $p_{mix}(t)$ as a mixture of these two Gaussian distributions by

$$p_{mix}(t)=\alpha p_B(t)+(1-\alpha)p_F(t) \qquad (4)$$

where α is determined by the portions of background and foreground in the image.

B. Local Thresholding Methods:

a. Niblack Algorithm

Niblack's algorithm calculates a pixel-wise threshold by sliding a rectangular window over the gray level image [4]. The computation of threshold is based on the local mean m and the standard deviation s of all the pixels in the window and is given by the following equation (5)

$$T_{Niblack}=m+k*s$$

$$T_{Niblack}=m+k\sqrt{\frac{1}{NP}\sum(p_i-m)^2}$$

$$=m+k\sqrt{\frac{\sum p_i^2}{NP}-m^2}=m+k\sqrt{B} \qquad (5)$$

where NP is the number of pixels in the gray image, m is the average value of the pixels p_i, and k is fixed to -0.2 by the authors. Advantage of Niblack is that it always identifies the text regions correctly as foreground but on the other hand tends to produce a large amount of binarization noise in non-text regions also.

C. Proposed Thresholding Algorithm

In many application of image processing, the gray levels of pixels belonging to the object are substantially different from the gray levels belonging to the background. Thresholding then becomes a simple but effective tool to separate objects from the background. There are many methods for thresholding. All this algorithms have their own metrics and defects at the same time. Maximum between-class square error should be a good method to select proper threshold on the condition that images must be processed on real time and the number of pixels in each class are close to each other. In nature, maximum classes square error belongs to Otsu method, which employs a criterion for maximizing the between-class variance of pixel intensity and can detect proper threshold. The antonym of Otsu is to pursue the maximum between-class variance, which can be viewed as an optimization problem [12]. Recently as it known from literature, Particle Swarm Optimization (PSO) algorithm is used to solve many of difficult problems in the field of pattern recognition [8]. Hence, we can employ PSO to deal with it.

a. PSO Algorithm

Let X and V denote the particle's position and its corresponding velocity in search space respectively. At iteration K, each particle i has its position defined by $X_i^k=(x_{i1},x_{i2},\ldots x_{in})$ and a velocity is defined by $V_i^k=(v_{i1},v_{i2},\ldots v_{in})$ in search space n. Velocity and position of each particle in next iterations can be calculated using following equation (6) and (7)

$$v_{ij}^{k+1}=wv_{ij}^k+c_1r_1\left(pbest_{ij}^k-x_{ij}^k\right)+c_2r_2\left(gbest_j^k-x_{ij}^k\right) \qquad (6)$$

$$x_{ij}^{k+1}=x_g^k+v_g^k \qquad (7)$$

where *k* is the current iteration number, *w* is inertia weight, v_{ij} is then updated velocity on the j^{th} dimension of the i^{th} particle, c_1 *and* c_2 are acceleration constants, c_1 and c_2 are positive constant parameters, usually $c_1=c_2=2$. r_1 and r_2 are the real numbers drawn from two uniform random sequences of U (0, 1).

The quality of solution is measured by the objective function. The fitness function f (t) of each particle can be formulated as equation (8).

$$f(t)=\omega_0(t)\times\omega_1(t)\times(\mu_0(t)-\mu_1(t))^2 \qquad (8)$$

Such that t is a gray level between 0 and 255 which can be obtained through the particle's position. $\omega_0(t)$ is amount of pixels whose gray value is lower than t, $\omega_1(t)$ is amount of pixels whose gray value is higher than t, $\mu_0(t)$ is the means of pixels with gray value less than t, $\mu_1(t)$ is the means of pixels with gray value more than t.

The algorithm starts by generating randomly initial population of the PSO. Every particle is initialized with locations and velocities. These locations composed the initial solutions for the optimal threshold. Next the particle's fitness is calculated by the equation (8). The algorithm keeps an updated version of two special variables through out its execution, *global best (gbest)* position and *local best (pbest)* position. It compares different positions of a particle with its current position, in order to determine the local best position for every particle. The procedure of the proposed PSO algorithm is described as follows:

Step 1: Initialize N particles with random positions $x_1, x_2, \ldots x_N$ according to Eq. (6) and velocities V_i i = 1, 2,...N.
Step 2: Evaluate each particle according to Eq. (8).
Step 3: Update individual and global best positions. if $f(pbest_i) < f(x_i)$, then $pbest_i = x_i$, and search for the maximum value f_{max} among $f(pbest_i)$, If max $f(gbest) < f_{max}$, then $gbest = x_{max}$, x_{max} is the particle associated with f_{max}.
Step 4. Update velocity: update the *i*th particle velocity using the Eq. (7) restricted by maximum and minimum threshold v_{max} and v_{min}.
Step 5. Update Position: update the i^{th} particle position using Eq. (6) and (7).
Step 6. Repeat step 2 to 5 until a given maximum number of iterations is achieved or the optimal solution so far has not been improved for a given number of iteration.

III. TEST AND RESULTS

The images are binarized by each of the binarization algorithm described in the above section. In order to measure the performance of these algorithms various evaluation metrics such as Precision, Recall, F-measure [7] and traditional measures of image quality description are used. Specifically, we used the square error (MSE), the signal to noise ratio (SNR) and the peak signal to noise ratio (PSNR) [5].

The true positives are those pixels that were black in the ground truth image and are still black in the binarized image. The false positives were white in the ground truth image and black in the binarized image. The false negatives are black in the ground truth image, but white in the binarized image. From these counts the statistics of the following are calculated [2].

$$\mathrm{Recall} = \frac{TruePositive}{FalseNegatives + TruePositives} \quad (9)$$

$$Percision = \frac{TruePositive}{FalsePositives + TruePositives} \quad (10)$$

These are combined into F-measure using the following equation

$$F - measure = \frac{2.\mathrm{Re}call.Percision}{\mathrm{Re}call + Percision} \quad (11)$$

Let x (i, j) represent the value of the i^{th} row and j^{th} column pixel in the original image x and let y (i, j) represent the value of the corresponding pixel in the output image y. The local error is $e(i,j) = x(i,j) - y(i,j)$ and the total square error rate will be as in equation (12).

$$MSE = \frac{\sum_i \sum_j e(i,j)^2}{MxN} \quad (12)$$

SNR is defined as the ration of average signal power to average noise power and for an MxN image is

$$SNR(db) = 10\log_{10}\left(\frac{\sum_{i,j} x(i,j)^2}{\sum_{i,j}(x(i,j) - y(i,j))^2}\right) \quad (13)$$

for $0 \le i \le M-1$ and $0 \le j \le N-1$. PSNR is defined as the ratio of peak signal power to average noise power

$$PSNR(db) = 10\log_{10}\left(\frac{D^2 MN}{\sum_{i,j}(x(i,j) - y(i,j))^2}\right) \quad (14)$$

for $0 \le i \le M-1$ and $0 \le j \le N-1$, where D is the maximum peak-to-peak swing of the signal (255 for 8-bit images). We assume that the noise $x(i,j) - y(i,j)$ is uncorrelated with the signal.

Table 1 shows the value of the evaluation metrics obtained for the proposed method and the other existing methods. From the table, we can observe that the performance of the proposed PSO method is better than the other methods. Fig 2. Shows the original images and the resultant binarized image obtained by various methods.

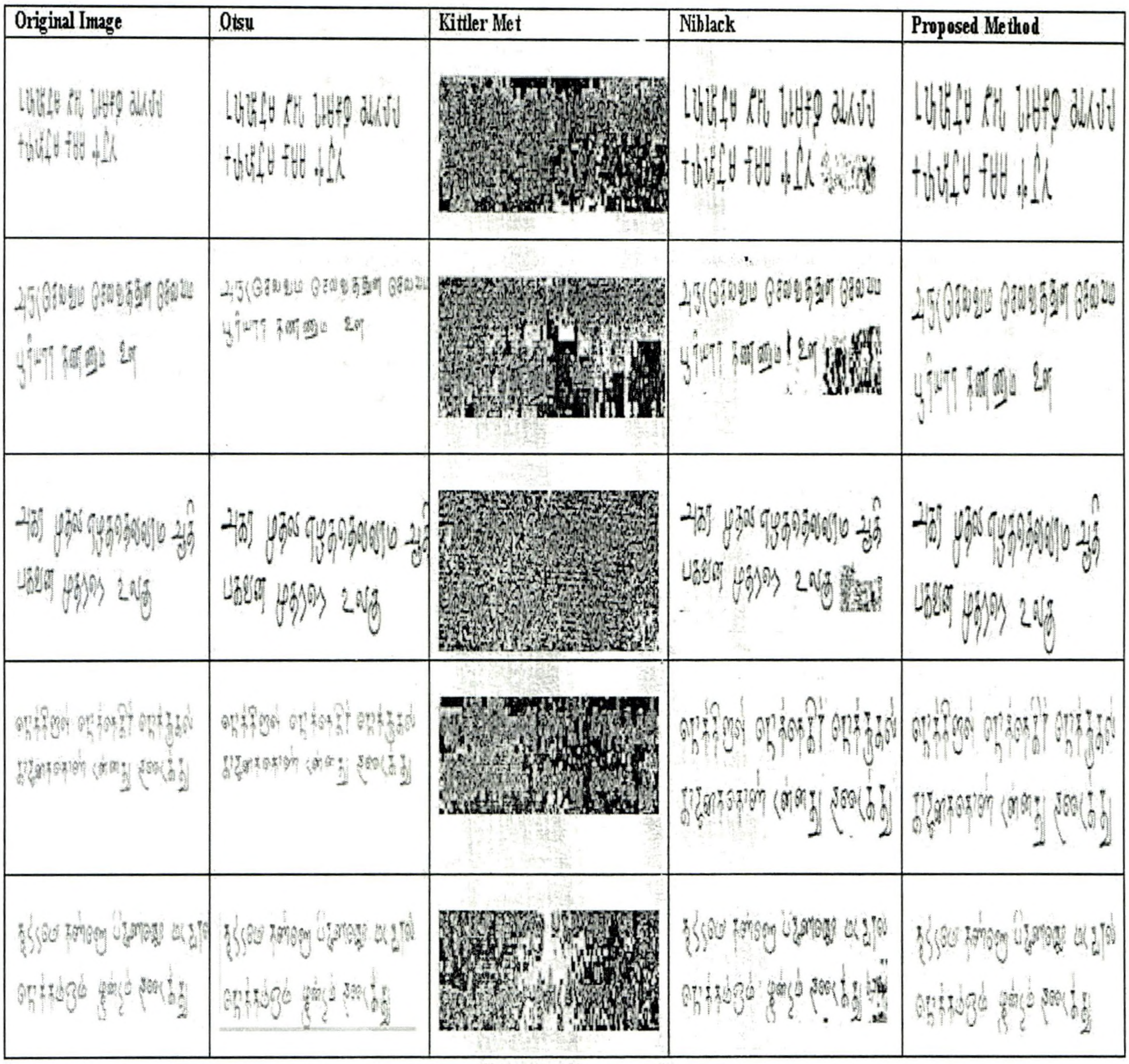


Figure 2: Original Image and Binarized Images using various algorithms

Table 1: The value of evaluation metrics for every binarization technique concerning the image of Figure 1

Performance Metrics	Otsu Algorithm	Niblack Algorithm	Kittler Met Algorithm	Proposed Method
PSNR	18.878	16.516	7.61	**19.322**
SNR	30.688	30.86	33.668	30.72
MSE	28.988	37.86	107.02	**27.94**
Precision	0.4726	0.5816	0.7845	0.5120
Recall	0.4513	0.3170	0.0633	0.4818
F-Measure	46.1731	40.4879	11.7063	**49.6418**
Sensitivity	0.4513	0.3170	0.0633	0.4818
Specificity	0.9721	0.9764	0.9684	0.9731

IV. CONCLUSION

This paper has presented comparison of various binarization algorithms by measuring their performance by evaluation metrics. In this paper, a new binarization method based on Particle Swarm Optimization (PSO) is proposed. A total of four different binarization methods such as Otsu, Niblack and Kittler Met along with proposed PSO method are considered and evaluated by evaluation metrics. From the experimental result, we can infer that proposed PSO method shows good result when compared with other methods, since their PSNR, SNR measures are higher and the MSE is lower. The higher value of PSNR means that the quality of the binarized image is better. According to the results, Proposed PSO method had the best overall performance with F-measures of 49.6418 which is higher when compared to other methods.

REFERENCES

[1] Chun Che Fung and Rapeeporn Chamchong, "A Review of Evaluation of Optimal Binarization Technique for Character Segmentation in Historical Manuscripts", Third International Conference on Knowledge Discovery and Data Mining, 2010.

[2] Junker.M, R. Hoch, "On the Evaluation of Document Analysis Components by Recall, Precision, and Accuracy", Proceedings of 5th ICDAR, India, pp. 713-716, 1999.

[3] Kennedy. J, and R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE conference on neural networks – ICNN"95, Perth, Australia, pp. 1942-1948, 1995.

[4] Khurram Khurshid, Imran Siddiqi, Claudie Faure, Nicole Vincent , "Comparison of Niblack inspired Binarization methods for ancient documents", 16th International conference on Document Recognition and Retrieval, 2009

[5] Kite, T.D., Evans, B.L., Daamera-Venkata, N., and Bovil, A.C.: "Image Quality Assessment Based on a Degradation Model"; in IEEE Trans. Image Processing, vol.9, pp.909-922, 2000.

[6] Kittler.J and J.Illingworth, "Minimum Error Thresholding", Pattern Recognition, Vol 19, No.1, pp. 41-47,1986

[7] Ntirogiannis .K, B. Gatos and I. Pratikakis, "An Objective Evaluation Methodology for Document Image Binarization Techniques", the Eighth IAPR Workshop on Document Analysis Systems, IEEE Transactions, 2008.

[8] Oliveira .S.L., S. A. Britto, and R. Sabourin, "Optimizing Class-Related Thresholds with Particle Swarm Optimization", Proceedings of International Joint Conference on Neural Networks, IEEE, Montreal, Canada, July 31 - August 4, pp. 1511-1516, 2005.

[9] Otsu, N., "A threshold selection method from gray level histograms," IEEE Trans. Syst. Man Cybern. 9, pp. 62–66, 1979.

[10] Pavlos Stathis, Ergina Kavallieratou, Nikos Papamarkos, "An Evaluation Technique for Binarization Algorithms", Journal of Universal Computer Science, vol. 14, no. 18, 2008.

[11] Russell Eberhart, James Kennedy, "A New Optimizer Using Particle Swarm Theory", Proceedings of Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995.

[12] Ye zhiwei Chen, hongwei Liu, Wei Zhang jinping," Automatic threshold selection based on Particle Swarm Optimization algorithm", International Conference on Intelligent Computation Technology and Automation, IEEE, 2008.

Dr. P. Subashini, Associate Professor, Dept. of Computer Science, Avinashilingam Deemed University has 18 years of teaching and research experience. Her research has spanned a large number of disciplines like Image analysis, Pattern recognition, neural networks, and applications to Digital Image processing. Under her supervision she has seven research project of worth one crore from various funding agencies like DRDO, DST and UGC.

N. Sridevi M.C.A, Ph.D Research Scholar, Dept. of Computer Science, Avinashilingam University has 3 years of teaching experience and 1 year of Industrial experience. Doing her research work in Pattern recognition. Area of Specialization: Image Processing, Pattern recognition.