Use of Splines in Handwritten Character Recognition

Sunil Kumar
Research Scholar, Singhania University
Rajasthan,
India

Satish Kumar
Electronics and Communication
NCR College,
Bahadurgarh, India

Gopinath S
ABB Global Industries,
Banglore, India

Rajesh Chhikara
Electronics and Communication
MIT Mundka, Delhi, India

Abstract— **Handwritten Character Recognition is software used to identify the handwritten characters and receive and interpret intelligible handwritten input from sources such as manuscript documents. The recent past several years has seen the development of many systems which are able to simulate the human brain actions. Among the many, the neural networks and the artificial intelligence are the most two important paradigms used. In this paper we propose a new algorithm for recognition of handwritten texts based on the spline function and neural network is proposed. In this approach the converse order of the handwritten character structure task is used to recognize the character. The spline function and the steepest descent methods are applied on the optimal notes to interpolate and approximate character shape. The sampled data of the handwritten text are used to obtain these optimal notes. Each character model is constructed by training the sequence of optimal notes using the neural network. Lastly the unknown input character is compared by all characters models to get the similitude scores.**

Index Terms—Artificial Neural Network, Back propagation algorithm, Optimal knots, Splines.

I. INTRODUCTION

Handwriting recognition is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface.

There has been a lot of research on handwritten character recognition in recent years, resulting in a number of proposed pattern recognition techniques. One such method uses Mahalanobis generalized distance of a feature vector calculated from a character image [6, 4]. A pattern recognition approach using statistical processing based on Bayes' theorem was proposed by Graham for distinguishing spam (junk) mail [3, 2]. Handwritten character recognition method using the Bayesian filter algorithm was proposed by [8]. Techniques using a machine-learning approach such as a neural network (NN) and a support vector machine (SVM) are also well known [1, 5, 7].

Good recognition rates are achieved for character or numeral recognition, where the number of classes is rather small. But as the number of classes increases, as for example in isolated word recognition, the recognition rates drop significantly. An even more difficult task is the recognition of general handwritten text lines or sentences. Here, the lexicon usually contains a huge amount of word classes and the correct number of words in the image is unknown in advance, which leads to additional errors. In this field, recognition rates between 50% and 80% are reported in literature, depending on the experimental setup [9, 10, 11, 12, 13]. A novel approach of on-line handwritten character recognition using natural spline is discussed in [14].

The various methods for character recognition have already been published. But the method presented here is advanced than those methods since cursive handwriting can be recognized with the help of a combination of spline function and artificial neural networks, this becomes the primary advantage of the method over other existing methods.

The purpose of our proposed method is to recognize characters using spline function. The continuous image of the character acquired is converted into discrete image using Digital Image Processing Techniques such as thinning, image filtering, rotate and converting a colored image into black white.

The Spline curves of all the characters were obtained together with their error function. The Spline matrices obtained were then used as inputs to the Artificial Neural Network (ANN). And the outputs of the network were character matrices.

ANN was then trained using Multilayer Back propagation algorithm, to correspond various spline curve to their respective characters. Hence the character can be recognized.

The flowchart in "fig.1" shows the various steps used in this method.

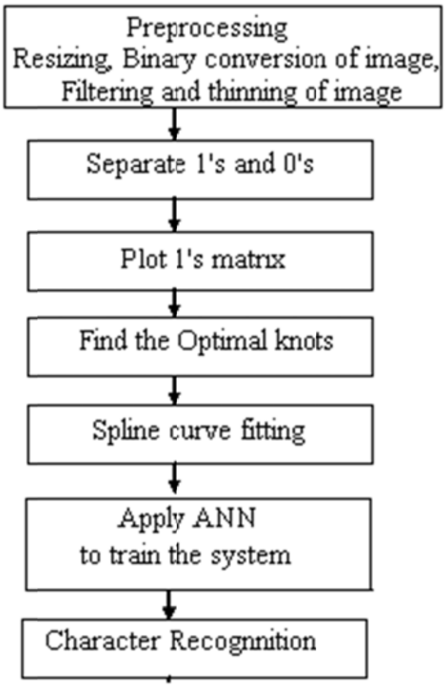


Figure 1. Flowchart representation of the various steps

Section II of this paper discusses the spline functions. Optimal knot detection process is explained in Section III. Result and Conclusion is discussed in Section IV and Section V.

II. Spline function

Splines are a class of functions that are used in data interpolation and/or smoothing. In case of interpolation the spline functions are determined as the minimizers of suitable measures of roughness subject to the interpolation constraints. In case of smoothing the splines may be viewed as generalizations of interpolation splines where the functions are determined to minimize a weighted combination of the average squared approximation error over observed data and the roughness measure.

A spline (S) is a piecewise polynomial function. This function maps the interval [a,b] to a set R of real numbers.

$$S : [a, b] \rightarrow R \quad (1)$$

We want S to be piecewise defined. To accomplish this, let the interval [a, b] be covered by k ordered, disjoint subintervals,

$$[xi, xi+1], 0 \leq i \leq k-1 \quad (2)$$

$$[a, b] = x0 \cup x1 \cup \ldots \cup xk-1 \quad (3)$$

On each of these k "pieces" of [a,b], we want to define a polynomial, call it Pi.

$$Pi : [xi, xi+1] \rightarrow R. \quad (4)$$

On the ith subinterval of [a,b], S is defined by Pi,

$$S(x) = p0(x), x0 \leq x1$$

$$S(x) = p1(x), x1 \leq x< x2$$

$$S(x) = pk-2(x), x1k-2 \leq x\leq xk-1 \quad (5)$$

The given k points xi are called knots. The vector x = (x0, x1….xk-1) is called a knot vector for the spline.

If S € Cxi in a neighborhood of xi, then the spline is said to be of smoothness (at least) Cxi at xi. That is, at xi the two pieces Pi-1 and Pi share common derivative values from the derivative of order 0 (the function value) up through the derivative of order ri (in other words, the two adjacent polynomial pieces connect with loss of smoothness of at most n - ri).

A vector r = (r1, r2,…., rk-2) such that the spline has smoothness Cxi at xi for 0 < i < k - 1 is called a smoothness vector for the spline.

A common spline is the natural cubic spline of degree 3 with continuity C2. The second derivatives of the spline polynomials are set equal to zero at the endpoints of the interval of interpolation

S"(a) \, = S"(b) = 0.

This force the boundaries of the spline curve to be inflection points. This twice differentiable continuity and its coupling of the system of equations dictating all other cubic fits in the spline generate the end result.

III. Optimal knots detection process

The first the second order derivation and low-pass filtering is applied on the character in order to detect the optimal knots. The derivated and filtered signal is divided by zero-crossing points. As initial optimal knots we considered the maximum absolute value point in each segment including the start and final points. The Slalom method and the steepest descent methods are used for the adjustment of the optimal knots.

The Slalom method must satisfy the following two conditions.

The difference between the spline function s(x) and a given fi (i=1 ... M) must be smaller than a previously determined value, δ.

$$\left| s(x) - fi \mid \leq \delta \right| \quad i=1,2 ,...M, \quad (6)$$

The Spline function s(x) must be a smooth function, that does not need to cross over for every given points fi (i=l . . .M).

To satisfy the two conditions given above, an error function J[s] must be minimized. We can define the first derivative J'[s] of the error function J[s] as eq. (7).

$$J^{'}[s] = \int (\frac{d^2}{dx^2} s(x))^2 dx + \propto \sum_{i=0}^{M} (s(x_i) - f_i)^2 dx \quad (7)$$

and the first and second derivative of s(x) correspondent to i+1th knot in discrete way, is written as:

$$s'_{i+1} = \frac{s_{i+1} - s_i}{x_{i+1} - x_i} = \frac{s_{i+1} - s_i}{\Delta} \qquad (8)$$

$$s''_{i+1} = \frac{1}{\Delta}\left(\frac{s_{i+1} - s_i}{\Delta} - \frac{s_i - s_{i-1}}{\Delta}\right) = \frac{s_{i+1} - 2s_i + s_{i-1}}{\Delta^2} \qquad (9)$$

where A is the interval between the i-th and the i+l-th knots. Supposing that intervals between two consecutive knots is equal to 1, we have

$$s''_{i+1} = s_{i+1} - 2s_i + s_{i-1} \qquad (10)$$

Next equation (2) can be rewrite using as,

$$J'[s] = \sum_{j=2}^{N-1}(s_{j+1} - 2s_j + s_{j-1})^2 + \propto \sum_{i=0}^{M}(s_{ji} - f_i)^2 \qquad (11)$$

where N is the samples number and M is the number of knots,

s_j is the i-th value from Spline function s(.) and s" is the equivalent value of s(.) of the i-th knot.

The minimization problem of J'[s] can be solve as follows.

$$\frac{\partial J'}{\partial s_k} = 0 \qquad k = 1 \ldots \ldots N \qquad (12)$$

Substituting eq. (6) into eq. (7), we can get,

$$\frac{\partial J'}{\partial s_1} = 2(s_1 - 2s_2 + s_3) + 2\alpha\,\partial_{1\Omega}(s_1 - f_1) = 0$$

$$\frac{\partial J'}{\partial s_2} = 2(-2s_1 + 5s_2 - 4s_3 + s_4) + 2\alpha\,\partial_{2\Omega}(s_2 - f_2) = 0$$

$$\frac{\partial J'}{\partial s_{N-1}} = 2(-2s_{N-3} + 5s_{N-2} - 4s_{N-1} + s_N) + 2\alpha\,\partial_{(N-1)\Omega}(s_{N-1} - f_{N-1}) = 0$$

$$\frac{\partial J'}{\partial s_N} = 2(s_{N-2} - 2s_{N-1} + s_N) + 2\alpha\,\partial_{N\Omega}(s_N - f_N) = 0 \qquad (13)$$

to get the gk, for k = 1, 2, ..., N, the following lineal equations must be resolved

$$\begin{bmatrix} 1 & -2 & 1 & & & & \\ -2 & 5 & -4 & 1 & & & \\ . & . & . & . & . & & \\ . & . & . & . & . & . & \\ & & 1 & -4 & 5 & -2 & \\ & & & 1 & -2 & 1 & \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ . \\ . \\ s_N \\ s_N \end{bmatrix} = \begin{bmatrix} 2\alpha\partial_{1\Omega}(s_1 - f_1) \\ 2\alpha\partial_{2\Omega}(s_2 - f_2) \\ . \\ . \\ 2\alpha\partial_{(N-1)\Omega}(s_{N-1} - f_{N-1}) \\ 2\alpha\partial_{N\Omega}(s_N - f_N) \end{bmatrix} \qquad (14)$$

Where Ω is the sampling space and δj,Ω satisfies,

$$\partial_{j\Omega} = \begin{cases} 0, & j \notin \Omega \\ 1, & j \in \Omega \end{cases} \qquad (15)$$

δj,Ω = 0, when j is different from any knot position and δj,Ω = 1, when j corresponds with the some knot position.

After the Slalom method is applied and optimal knots are obtained, the error E, can be got by eq. (16).

$$E = \frac{1}{N}\sum_{j=1}^{N}(f(x_i) - s(x_i))^2 \qquad (16)$$

To minimize the error E, the steepest descent method is applied. The steepest descent method is the simplest gradient methods used to minimize a given error function. Then the optimal knot's position update is given by eq. (17).

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k) = x_k - \lambda_k s(x_k) \qquad (17)$$

After the adjustment of all knots position is performed, we analyze the distance of all consecutive knots. Next if the distance between some two consecutive knots is larger than a determinated threshold value, a new knot is added in the center position of the two knots.

IV. Results

The figure 2 shows the original character "A" and figure 3 shows the same character after preprocessing. In figure 4 the one window represents the spline corresponding to "A" and the other window shows the error in spline. Figure 5 to 8 represents the ANN output for different Epochs. The different spline coefficients are shown in figure 9. Figure 10 and figure 11 shows the splines corresponding to other sample characters.

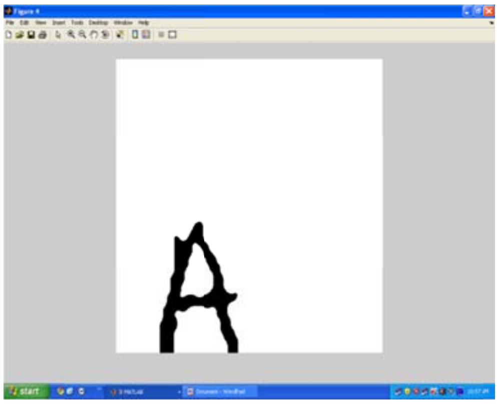


Figure 2. Offline Image of character 'A'

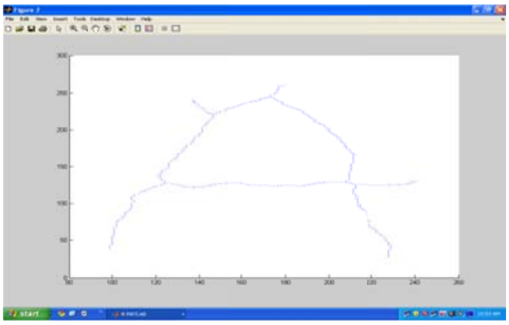


Figure 3. Filtered thin image of the character representing nodes

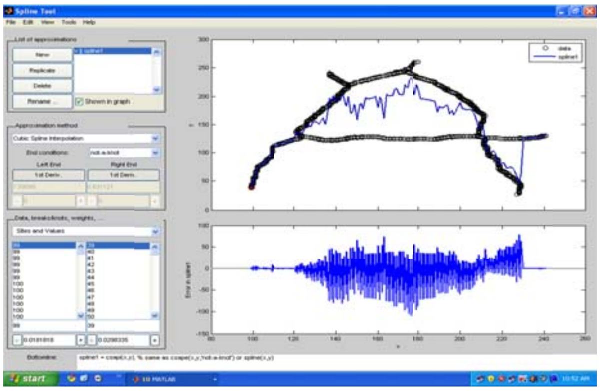


Figure 4. Spline curve for Character ‘A’ and Error in spline

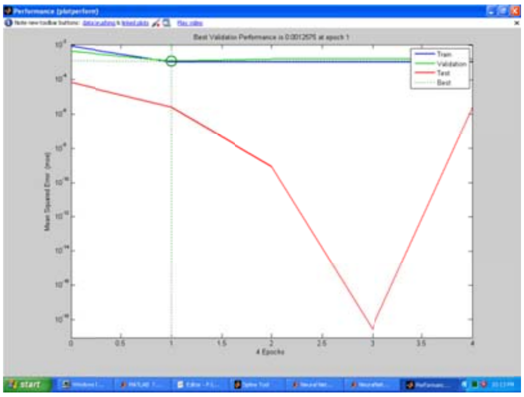


Figure 5.ANN output for Epoch 1

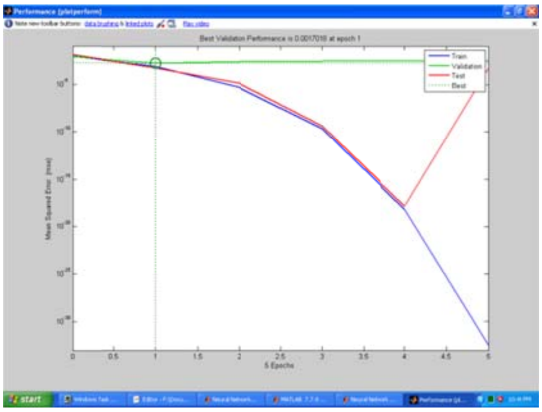


Figure 6. ANN output for epoch 2

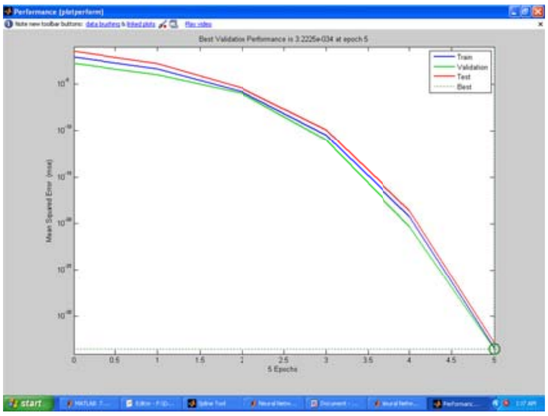


Figure 7:ANN output for epoch 5

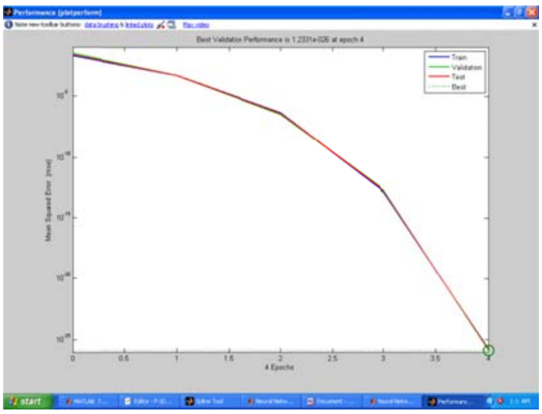
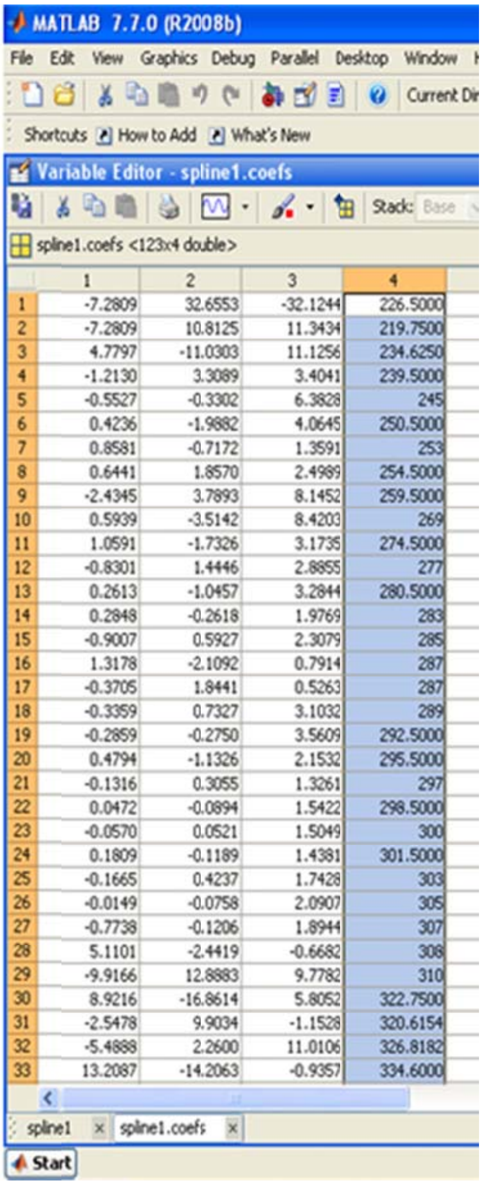


Figure 8 : ANN output for epoch 4



	1	2	3	4
1	-7.2809	32.6553	-32.1244	226.5000
2	-7.2809	10.8125	11.3434	219.7500
3	4.7797	-11.0303	11.1256	234.6250
4	-1.2130	3.3089	3.4041	239.5000
5	-0.5527	-0.3302	6.3828	245
6	0.4236	-1.9882	4.0645	250.5000
7	0.8581	-0.7172	1.3591	253
8	0.6441	1.8570	2.4989	254.5000
9	-2.4345	3.7893	8.1452	259.5000
10	0.5939	-3.5142	8.4203	269
11	1.0591	-1.7326	3.1735	274.5000
12	-0.8301	1.4446	2.8855	277
13	0.2613	-1.0457	3.2844	280.5000
14	0.2848	-0.2618	1.9769	283
15	-0.9007	0.5927	2.3079	285
16	1.3178	-2.1092	0.7914	287
17	-0.3705	1.8441	0.5263	287
18	-0.3359	0.7327	3.1032	289
19	-0.2859	-0.2750	3.5609	292.5000
20	0.4794	-1.1326	2.1532	295.5000
21	-0.1316	0.3055	1.3261	297
22	0.0472	-0.0894	1.5422	298.5000
23	-0.0570	0.0521	1.5049	300
24	0.1809	-0.1189	1.4381	301.5000
25	-0.1665	0.4237	1.7428	303
26	-0.0149	-0.0758	2.0907	305
27	-0.7738	-0.1206	1.8944	307
28	5.1101	-2.4419	-0.6682	308
29	-9.9166	12.8883	9.7782	310
30	8.9216	-16.8614	5.8052	322.7500
31	-2.5478	9.9034	-1.1528	320.6154
32	-5.4888	2.2600	11.0106	326.8182
33	13.2087	-14.2063	-0.9357	334.6000

Figure 9: Coefficients of Splines

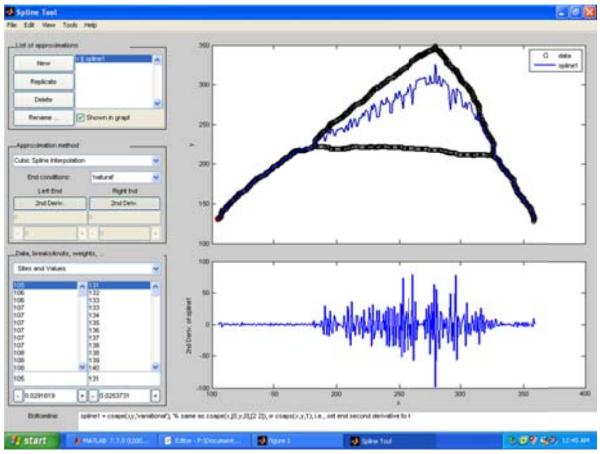


Figure 10. Spline for another sample of "A"

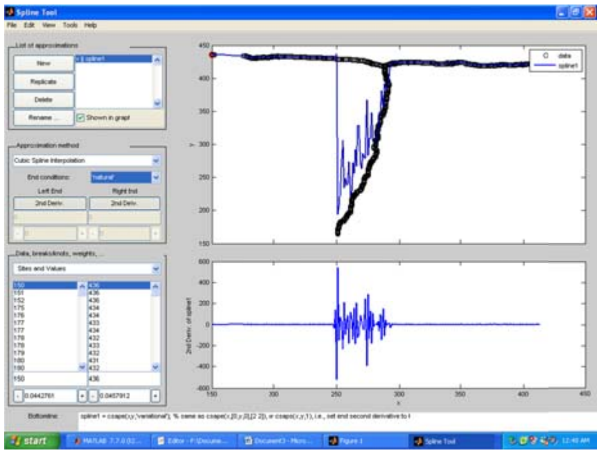


Figure 10. Spline for sample character "T"

The proposed method is applied to a data base of 390 different English alphabet characters(all are in upper case). The overall recognition rate by this method is about 91.5%. The recognition rate of the methods like Method based on line inclination and relative position of line(Namboodri System) (95.5%) and PBD Template method(92.3) achieve slightly higher recognition rates but that is at the expense of complexity as well as lower processing speed.

As far as individual characters are concern, the characters "A,C,E,F,L,O,P,Q,V,W,Y" achieved a recognition rate as high as 96% where as the alpha bets like "S,D,H,K,M" have the lowest recognition rate (82.5%). The rate of recognition of the remaining other characters "G,T,U,B,I,J,N,R,X,Z" lies somewhere in between 85% to 90%.

V. CONCLUSION

In this paper, a new method for feature extraction from the handwritten character is proposed. The proposed method, successfully implemented gives an efficiency of around 91.5%. So far we have been able to produce a discrete image of the character with distinct nodes. The redundant nodes were eliminated.

The character hence obtained is then interpolated using Spline functions to produce a character curve. Spline is a piecewise polynomial function, hence piecewise curves are obtained.

Artificial Neural Network is then implemented and trained to produce various character matrix, corresponding to their respective Spline matrices.

REFERENCES

[1] D. A. Forsyth and J. Ponce. Computer Vision A Modern Approach. Prentice Hall, 2003.

[2] P. Graham. Better bayesian filtering. http://www.paulgraham. com/better.html.

[3] P. Graham. A plan for spam. http://www.paulgraham.com/ spam.html.

[4] Y. Hotta, S. Naoi, and M. Suwa “Improvement of numeral recognition using personal handwriting characteristics based on clustering” Journal of IEICE, J83-D-II(3):871–878, 2000.

[5] T. Ishii, Y. Waizumi, N. Kato, and Y. Nemoto “Recognition system for handwritten characters by alternative method using neural network” Journal of IEICE, J86-D-II(3):988–995, 2003.

[6] Y. Waizumi, N. Ebisawa, N. Kato, and Y. Nemoto “Handwritten character recognition using learning pattern generation by nonlinear normalization” Journal of IEICE, J86-DII(10):1391–1399, 2003.

[7] Y. Waizumi, N. Kato, and Y. Nemoto “High accuracy recognition for handwritten characters using multi layered perceptron with squared connections” Journal of IEICE, J83-DII(10):1969–1976, 2000.

[8] Nozomu Araki, Miyuki Okuzaki, Yasuo Konishi and Hiroyuki Ishigaki “A Statistical Approach for Handwritten Character Recognition Using Bayesian Filter” The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC'08), 2008 IEEE

[9] G. Kim, V. Govindaraju, and S. Srihari “Architecture for handwritten text recognition systems” In S.-W. Lee, editor, Advances in Handwriting Recognition, pages 163–172. World Scientific Publ. Co., 1999.

[10] A. Senior and A. Robinson “An off-line cursive handwriting recognition system” IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3):309–321, 1998.

[11] A. Vinciarelli, S. Bengio, and H. Bunke “Offline recognition of unconstrained handwritten texts using HMMs and statistical language models” IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):709–720, 2004.

[12] A. Vinciarelli and J. Luettin “Off-line cursive script recognition based on continuous density HMM” In 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands, pages 493– 498, 2000.

[13] M. Zimmermann and H. Bunke “Optimizing the integration of a statistical language model in HMM based offline handwriting text recognition” In 17th International Conference on Pattern Recognition, Cambridge, England, volume 2, pages 541 – 544, 2004.

[14] Makato Yasuhara “On-Line Handwritten Character Recognition” The 47th IEEE International Midwest Symposium on Circuit and Systems.