Chinese Chess Character Recognition using Direction Feature Extraction and Backpropagation

Seniman, Dedy Arisandi, Romi Fadillah Rahmat, William, Erna Budhiarti Nababan
Dept. of Information Technology
Faculty of Computer Science and Information Technology
University of Sumatera Utara
Medan, Indonesia
seniman@usu.ac.id, dedyarisandi@usu.ac.id, romi.fadillah@usu.ac.id, william_leolim @students.usu.ac.id, ernabrn@usu.ac.id

Abstract—**Backpropagation and Direction Feature Extraction (DFE) are proposed in this paper for Chinese chess character recognition. Backpropagation is a feed-forward neural network algorithm designed for learning by examples namely by calculating errors and updating weights in each epoch. DFE is a feature extraction method by iterating and calculating the directons surrounding each pixel in the image to obtain the features. In this research, Chinese chess characters are recognized to obtain the correct amount of each chess character in a package. Due to the complex contour, stroke and pattern of Chinese chess characters, Chinese chess characters are difficult to be recognized by new learners. Both Backpropagation and DFE performance are capable in recognizing Chinese chess characters with good accuracy of 98% for various sets and it is also robust from transition, brightness, image noise and rotation up to 60°.**

Keywords—image recognition; Chinese chess characters; Direction Feature Extraction (DFE); Neural Network; Backpropagation

I. Introduction

The optical character recognition was started from the recognition of machine printed digits and characters and then it was developed to the recognition of machine printed words [1]. Chinese chess has various font types and various characters. The availability of sufficient training patterns is an essential requirement for designing high performance classifiers for character recognition [2]. Chinese character uniqueness relies on its stroke-shape. The parts defining stroke-shape, are mainly end parts and bend parts because writing-direction, pressure or pen posture often change significantly in these parts [3]. Therefore, this research focuses on recognizing Chinese chess character and various sets were trained and tested.

In 2011, Jia et al. proposed Radial Harmonic Fourier Moments (RHFMs) to recognize Chinese chess up to 10° of rotational transition with testing result rated 100% slightly higher than its training result rated 99.14% [4]. However, This method was only tested on one sample of Chinese chess set and one sample of artificial Chinese chess characters. Wen, in 2014, proposed feature comparisons technique to identify Chinese chess character with accuracy rate of 100%, resistance to 20% of noise and. 40° of incline attack [5]. Multilayer perceptron and Feature extraction techniques were also implemented to recognize cursive and non-aphabet characters, DevNagari characters, by Aurora et al with the accuracy rate of 92.8%. [6]. The proposed method is not rotational variant. MQDF and LVQ were also proposed to recognize Japanese handwritten characters with 97.23% accuracy without rotational variant by Velek [7].

Image enhancement is the task of applying certain alterations to an input image like as to obtain a more visually pleasing image [8]. Several pre-processing methods are used to enhance stable feature results. DFE outcomes unique features for each different rotated characters. Backpropagation allows neural network to calculate errors in each of its layer and update its weight in each epoch. A very important features of neural networks is their adaptive nature, where learning by example replaces progamming in solving problems [9]. The parameters in neural networks are: number of neurons in input layers, number of neurons in output layers, number of hidden layers, number of neurons in each hidden layer, epoch, learning rate and momentum rate [10].

In this research, Backpropagation and DFE are able to handle various data sets with various real-world Chinese chess sets and various artificial Chinese chess characters with rotational, transition and noise issues. The rest of this paper is organized as follows. General architecture of proposed methods are described in Section II. The result and discussion of this research are in Section III. Finally, Section IV provides conclusion of this paper and the future research for this paper.

II. GENERAL ARCHITECTURE

The general acrchitecture of proposed methods in this research is shown in Fig. 1. Each step will be explained in details on the next sections.

A. Dataset

Datasets used in this research are collected from various types of real-world Chinese chess sets. The first dataset is for real-world Chinese chess with fixed rotational degree were taken and applied to five sets in total of 1600 images for training and 400 images for testing with 10° random rotation transition. Secondly, Artificial datasets for Chinese chess were also taken

978-1-5090-5671-2/16/$31.00 ©2016 IEEE

in group of five different font types which are *Gu Cheng, DFB*, *Fang Song*, *Sim Sum* and *VTMei*. The total training images are 1600 images which were taken from real-world Chinese chess images and were tested againts 400 artificial images with 10° random rotational transition. The third datasets with higher rotational degree transition were taken from real-world Chinese chess. Training datasets were originated from real-world Chinese chess sets and programmatically generated from real-world Chinese chess for each 5° in 60°-120° rotational transition of 1040 images and tested to live images using HD Webcam in random degrees between 60° and 120° for 400 images.

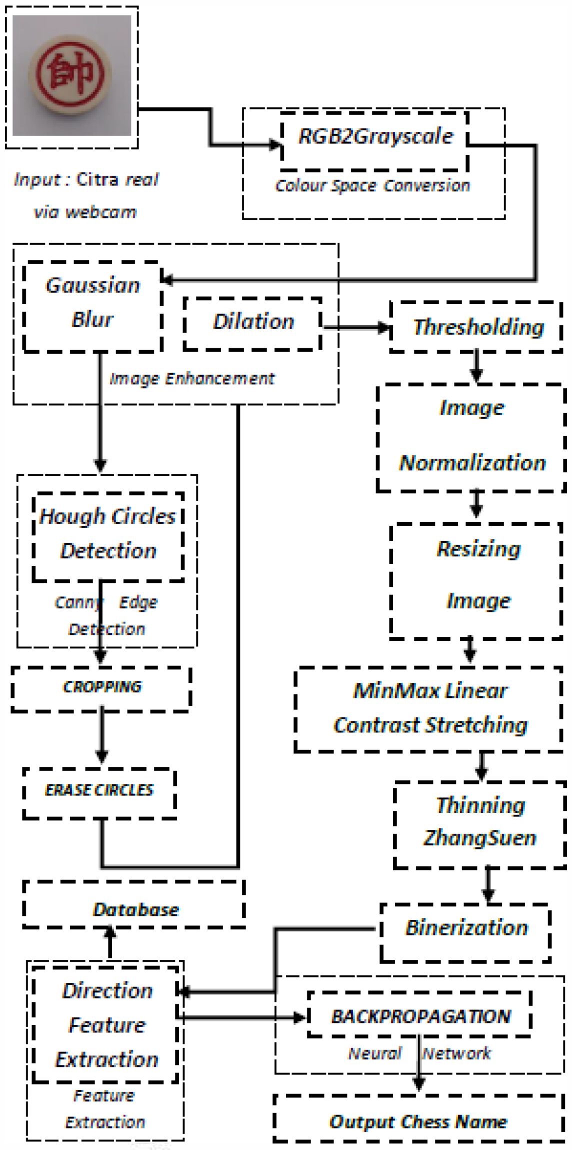


Fig. 1. General architecture of proposed methods

B. *Pre-processing*

Before extracting features from the input images, several pre-processing methods has to be executed to obtain the proper image's matric.

1) *Colour Space Conversion*

The RGB2GRAY converts RGB to grayscale image by eliminating the hue and saturation information while retaining the luminance [11]. Grayscale image is sized into 8-bit with the intensity ranging from 0 to 255. Converting RGB to Grayscale is shown in Equation 1 below.

$$Y = 0.299 \,.\, R + 0.588 \,.\, G + 0.114 \,.\, B \qquad (1)$$

2) *Image Enhancement*

Image enhancement to noise in grayscale images is undisputed. Gaussian Blur is applied to smooth its image pixels. The kernel size used is 9x9. Gaussian Blur is shown in Equation 2.

$$G(x, \sigma^2) = a \,.\, e^{\frac{-x^2}{2\sigma^2}} \qquad (2)$$

Dilation and erosion are basic morphological processing operations [12]. Dilation is used to add within gaps of the missing pixels around the foreground contour. Dilation operation is shown in Equation 3.

$$X \oplus A = X + a = \{x + a: x \in X \,\&\, a \in A\} \qquad (3)$$

X is the original image and *A* is the structural elements. The set of the images and structural elements are variabled into *x* and *a*. Various iterations in dilation enhance the image quality for each chess set.

3) *Canny Edge Detection*

Edge detection is capable to reduce the data amount to be processed and provide structural element of its detected shapes for further process [13]. Circle elements (x, y, Radius) are obtained for removing ring projection on the chess by applying Hough Circle Detection.

4) *Cropping*

The image is cropped to fixed size of 300x300 pixels based on obtained structural circular elements. Starting point of the image is calculated from the centre of the circle. The starting point is shown as (x', y') in both Equation 4 and 5.

$$x' = (\text{circle width} / 2) - x \qquad (4)$$

$$y' = (\text{circle height} / 2) - y \qquad (5)$$

Removing ring projection from marked circular region in the RGB image needs changing its grayscale pixel into 255. The image is later on dilated for the second time.

5) *Thresholding*

Grayscale image contains in various intensity on each of its pixels which threshold seperates its values into binary values. Equation of threshold function is shown in Equation 6.

$$b(x,y) = \begin{cases} 1 & if\ I(x,y) > \ T(x,y) \\ 0 & otherwise \end{cases} \qquad (6)$$

Thresholded image is presented by *b()* with its x and y grid. If threshold value of *T()* in each x and y position is smaller than the image pixels, the binary value is 1 otherwise it is 0.

6) *Image Normalization*

In this research, trajectory normalization is applied by raster-checking on top, bottom, left and right sides of the image. Once these positions are obtained, the image is then boxed out depending on the size of the characters.

7) *Resizing*

Resizing all images into 24 x 24 pixels eases later process. Smoothing and interpolation will be done to the resized image for better image quality.

8) MinMax Linear Contrast Stretching

Min Max Linear Contrast Streching overcomes blur pixels caused by resizing into more obvious pattern. Equation 7 shows the MinMax Linear Contrast Stretching.

$$g(x,y) = \frac{(f(x,y) - \min)}{(\max - min)} \qquad (7)$$

9) Thinning

Image with thickness more than 1 pixel results unstable feature. Thinning results foreground image into one-pixel thickness with consequence of shape and pattern reduction.

10)Binerization

Image binerization is the process of separation of pixel values into two groups, white as background and black as foreground [14]. Binerization is applied to generate binary matrix of its image as the input for DFE.

C. Direction Feature Extraction (DFE)

DFE focuses on determining direction values on each pixels and only processes on the foreground in the binary set. The direction values choosen in this research are 2, 3, 4 and 5.

Each pixel in the binary set is converted by orbiting into the pixel's neighbours surrounding the observed pixel. The neighbour pixels range only one pixel around the observed pixel. Table 1 describes each directions values to be converted. Meanwhile, Figure 2 shows the centre pixel and its neighbour pixels.

TABLE I
DIRECTION VALUES

Direction	**Value**	**Shape**
Vertical	2	│
Right Diagonal	3	╱
Horizontal	4	—
Left Diagonal	5	╲

Meanwhile, background value is 0 in the binary set. Changing binary values from the binary set into direction set is based on iterating sequence. The iterating sequence is clockwise starting from P2-P6, P3-P7, P4-P8 and P5-P1.

P1	P2	P3
P8	P0	P4
P7	P6	P5

Fig. 2. Observed Pixel (P0) and Neigbour Pixels

After the direction sets are obtained, extraction process begins. In this research, the direction set sizes in 24x24 matrix. DFE checks the direction sets by raster based on four directions. During its raster checking, transition value is stored by dividing the direction value with value of 10. The values stored based on its transition amount. Values found less than its transition amount in a row or column are then filled with 0.

Now, the four matrixs of four directions are obtained with the size of 4x24 for each. The following step is to normalize the four matrixs by averaging four rows into one row. The first transition of the four rows are quantified and divided by the number of transition used which is 4 and so on with the second, third and fourth transition. As the result, each matrix sized into 4x6.

In the end of direction feature extraction, the set of feature is obtained by attaching each normalized matrices into the feature matrix with the size of four normalized matrix. Further method of DFE is MDF which is modified DFE method used to recognize cursive characters in the reseach by Liu et al [15]. In this research, the direction feature sizes 6x16 which has 96 elements in the sets as the input for the neural network.

D. Backpropagation

Biologically inspired computing has been an active area of research for the past decades [16]. Unlike other algorithms that only solve specific problems using specific methods, machine learning such as neural network is able to develop to solve problems through learning.

Backpropagation is one of the most ideal neural network on its feed-forward capability. The backpropagation algorithm and its variants have been the backbone for training SLFNs with addictive hidden nodes [17]. A machine learning algorithm's generalization capability depends on the dataset [18].

Backpropagation mainly consists of three different layers which are input layer, hidden layer and output layer. The input layer is the layer which maintains numbers of nodes. These nodes represents the elements for the input which is the features extracted. The hidden layer is the bridge from the input layer to the output layer. By having hidden layer in its network, it allows the Backpropagation to feed-forward its network and check the error rates of each layers. The output layer consists of output nodes in which each node represents the desired classification result.

Backpropagation works in classifying the input layer into the correct classification of the output nodes. Backpropagation can be describes as follows:

1. Initialize weights that bridge layers into random float value from -0.5 to 0.5 in this research. Feature elements are set into each input nodes in the input layer. Then, it propagates along the bridge to the hidden layer.
2. Input nodes in this layer are represented by *i*. Once it is entering the hidden layer, hidden nodes calculate the current sum of the weight in the input-hidden bridge with each of the input nodes from the input layer. Hidden nodes in this layer are represented by *h*.

   $$Sum\ h = \sum(Input(i) * weight(i,h)) \qquad (8)$$

   Meanwhile, current sum of weight h for bias is multiplied by 1 with the weight in the last node of the input which is bias.

   $$Sum\ h\ bias = \sum(\ 1 * weight(i+1,h)) \qquad (9)$$

   The output of hidden layer is calculated into the sigmoid activation function and is then propagated to the next layer which is the output layer. Sigmoid activation function is described in Equation 10.

$$f(x) = \frac{1}{(1+e^{x})} \quad (10)$$

$$Output\ H(h) = f(Sum\ h + Sum\ h\ bias\ ) \quad (11)$$

3. Same process in the previous layers, output of hidden layer is then propagated along the bridge from hidden to output layer. Hidden nodes in this layer are represented by *h.*
4. Once it is entering the output layer, output nodes then calculate the current sum of the weight in the hidden-output bridge with each of the hidden nodes from the hidden layer. Output nodes in this later are represented by *o*.

$$Sum\ o = \sum(Hidden(h) * weight(h, o)) \quad (12)$$

Same step with the previous layer, current sum of weight *o* for bias is multiplied by 1 with the weight in the last node of the hidden which is bias.

$$Sum\ o\ bias = \sum(\ 1 * weight(h + 1, o)) \quad (13)$$

The output of output layer is calculated into the sigmoid activation function in Equation 10.

$$Output\ O(o) = f(Sum\ o + Sum\ o\ bias\ ) \quad (14)$$

The end of the feed forward algorithm in Backpropagation results the output of the neural network. Feed forward algorithm is explained from step1 until step 4.

5. The output nodes are then compared to the target nodes in order to obtain the error for each output nodes. In order to backpropagate and find the error, sigmoid function is then derived into derived sigmoid function shown in Equation 15.

$$f'(x) = f(x) * (1 - f(x)) \quad (15)$$

Error in output layer is calculated and shown in Equation 16

$$Error\ o = \left(target(o) - output(o)\right) * f'(output(0)) \quad (16)$$

6. Error in the hidden layer is then processed by backpropagate to hidden layer. Error in the output layer is then calculated with the feed-forward weight remaining in the bridge and it results the error in hidden layer. Equation 17 shows the error rate calculation.

$$Error\ h = Error\ o * f'(weight(h, o)) \quad (17)$$

7. Weights bridging input-hidden and hidden-output are updated after the error rate in each layers are obtained. Weight in hidden and output layer is updated by the learning rate, the error rate in output layer and node in each hidden layers. Equation 18 shows updating weight of hidden-output layer.

$$Weight(h, o) = \sum(Learn\ rate * Error\ o * Hidden(h)) \quad (18)$$

8. Weights bridging input and hidden layer is also updated by learning rate, error rate in hidden layer and node in each input layers. Equation 20 shows updateing weight of input-hidden layer.

$$\text{Weight}(i, h) = \sum(\text{Learn rate*Error h*Input(i)}) \quad (19)$$

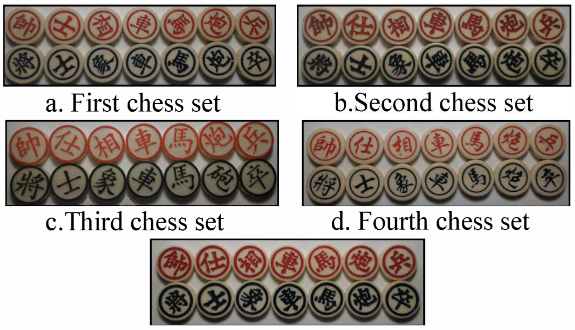
Process of neural network runs until it reaches the defined epoch of the training. Testing onto the network is only done until feed-forward process. Meanwhile, backpropagate process updates its network learning ability. Feed-forward is shown from step 1 till step 4 while backpropagate is shown from step 5 to step 8.

III. Result And Discussion

In this section, results from training and testing are obtained based on diversed font types and patterns, translation and rotational. Three experiments are tested to various real-world Chinese chess sets in a fixed rotational degree, various artificial Chinese chess characters and rotated real-world Chinese chess characters up to 60°.

A. Experiment on real-world Chinese chess sets in a fixed rotational degree

Five different sets were done for five times training and testing . The database consists of 16 Chinese chess characters and the total amount of 80 Chinese chess characters for all sets with different fonts. The fixed-rotational sample of real-world Chinese chess images are shown in Fig. 3.



a. First chess set

b.Second chess set

c.Third chess set

d. Fourth chess set

e. Fifth chess set

Fig. 3. Real-world Chinese chess set images for datasets.

The first experiment testing takes 320 training and 80 testing datasets. Due to excessive amount of traning sets and complex patterns, it requires 10.000 epoch for more desired classification results. The other testing datasets consist of images with random rotation from 85° to 95° with 400 images in random degree using webcam. The results of 90° Chinese chess classification based on different patterns are shown in Table II. Random rotational degree from 85° to 95° classification results are shown in Table III.

TABLE II
90° CHINESE CHESS TRAINING AND TESTING RESULTS

No.	Training Sets	Testing Sets	Training Results	Testing Results
1	II,III,IV,V	I	100%	87.5%
2	I,III,IV,V	II	100%	100%
3	I,II,IV,V	III	100%	100%
4	I,II,III,V	IV	100%	87.5%
5	I,II,III,IV	V	100%	100%
Overall			100%	95%

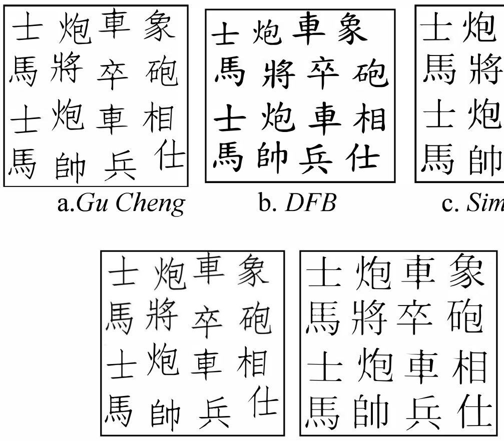
TABLE III
RANDOM 85° - 95° CHINESE CHESS TRAINING AND TESTING RESULTS

No.	Training Sets	Testing Sets	Training Results	Testing Results
1	I,II,III,IV	I	100%	97.92%
2	I,III,IV,V	II	100%	79.17%
3	I,II,IV,V	III	100%	95.84%
4	I,II,III,V	IV	100%	72.92%
5	I,II,III,IV	V	100%	83.34%
Overall			100%	85.84%

Having transition in its rotated degree, direction features change its value as the character's pattern rotates. This causes lower accuraccy rates in the rotated case which is 85.84%.

B. *Experiment on artificial Chinese chess characters*

The second experiment aims to test the neural network of Backpropagation to classify various digital characters as testing sets from real-world Chinese chess characters as training sets. Training iteration was 10.000 epochs. Training datasets are from the first experiment training datasets and its testing datasets are 80 artificial Chinese chess character images. The digital Chinese character sets are shown in Fig 4. The results of the second experiment are then shown in Table IV.


士炮車象馬將卒砲士炮車相馬帥兵仕
士炮車象馬將卒砲士炮車相馬帥兵仕
士炮車象馬將卒砲士炮車相馬帥兵仕
士炮車象馬將卒砲士炮車相馬帥兵仕
士炮車象馬將卒砲士炮車相馬帥兵仕

a.*Gu Cheng* b. *DFB* c. *Sim Sum*

d.*Fang Song* e.VTMei

Fig. 4. Digital Chinese characters

TABLE IV
ARTIFICIAL CHINESE CHESS CHARACTERS TRAINING AND TESTING RESULTS

No.	Training Sets	Testing Sets	Training Results	Testing Results
1	I,II,III,IV,V	*GuCheng*	100%	93.75%
1	I,II,III,IV,V	*DFB*	100%	87.5%
2	I,II,III,IV,V	*Fang Song*	100%	93.75%
3	I,II,III,IV,V	*Sim Sum*	100%	81.25%
4	I,II,III,IV,V	*VTMei*	100%	75%
Overall			100%	86.15%

The highest accuracy rate is achieved by *Gu Cheng* and *Fang Song*. Meanwhile, *Simsum* and *VTMei* show lower recognition due to irrelavant contours with the training sets. However, the proposed methods achieve 86.15% accuracy rate in recognizing digital Mandarin fonts which have so much irrelavant contours with the Chinese chess fonts.

C. *Experiment on rotated real-world Chinese chess characters up to 60°*

Chiness chess character recognition is performed in 60°-120° with rotational transition in 10° and 5°. The main aim of this research is to implement live-feed image recognition onto randomly-rotated Chinese chess. Training images are 1040 images and the testing images are captured randomly for 2000 images. Table V shows the classification result with 10° transition and Table VI shows the classification result with 5 °transition.

TABLE V
10° ROTATIONAL TRANSITION FROM 60° – 120° RANDOM TESTING DATASET RESULTS

No.	Training Sets (60°-120°) with 10° Transition	Testing Sets (65°-115°) with 10° Transition	Training Results	Testing Results
1	I	I	100%	86.61%
2	II	II	100%	86.72%
3	III	III	100%	91.65%
4	IV	IV	100%	90.48%
5	V	V	100%	88.10%
Overall			100%	88.71%

The training duration for 5° transition takes slighty twice longer than training the image with 10° transition. Larger epoch is given to both networks which is 200.000 epochs. However, larger leaning rate for excessive amount of traning data is advisable for good classification results. Lower recognition rate is on the fouth set. This is caused by complex contours and unique chaligraphic pattern by the fourth set.

TABLE VI

5° ROTATIONAL TRANSITION FROM 60° – 120° RANDOM TESTING DATASET RESULTS

No.	Training Sets (60°-120°) with 5° Transition	Testing Sets (65°-115°) with 5° Transition	Training Results	Testing Results
1	I	I	100%	98.75%
2	II	II	100%	98.75%
3	III	III	100%	100%
4	IV	IV	100%	95%
5	V	V	100%	100%
Overall			100%	98.50%

Higher recognition rates are obtained when lower transition is applied into the training images on the reason that DFE outputs closed features when both degrees are rotated close to each other. Even noise disturbance, transition and rotational transition up to 60° were applied in the testing process, the result shows good accuracy for rotated Chinese chess characters with 98.50% accuracy.

IV. CONCLUSION

In this paper, the proposed methods of DFE and Backpropagation are resistant to transition, noise, intensity changes and rotation up to 60°. It has been evaluated that the proposed methods obtain satisfactory traning results which are constantly 100% due to correct structural architecture and excessive learning rate in its network and testing results rate 98.5%. This research focused on learning complex patterns and countours on Chinese chess characters and uses various real-world Chinese chess sets up to five types of challigraphic fonts and five artificial Chinese chess characters. Mainly, more testing on various Chinese chess sets were tested in this paper than previous researches. The proposed methods has shown the potential to be implemented in pratical application to sort Chinese chess in industry. Future works involve further enhancement of the methods and its application in other areas.

ACKNOWLEGEMENT

The authors acknowledge that this paper is part of authors' outcome in University of Sumatera Utara institutional research as stated in agreement number 87/UN5.2.3.1/PPM/SP/2016.

REFERENCES

[1] B.H. Sumedha and D.S. Geeta, "Twelve Directional Feature Extraction for Handwritten English Character Recognition," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 2, 2013, pp. 39-42.

[2] O. Velek, C.L. Liu, S. Jaeger, and M. Nakagawa, "An Improved Approach to Generating Realistic Kanji Character Images from On-line Characters and its Benefit to Off-line Recognition Performance," *IEEE 16th International Conference on Pattern Recognition (ICPR)*, 2002, pp. 588 – 591.

[3] O. Velek and M. Nakagawa, "Vector-to-Image Transformation of Character Patterns for On-line and Off-line Recognition," *International Journal of Computer Processing of Oriental Languages*, vol. 15, 2002, pp. 187-209.

[4] W.K. Jia, Z.H. Gang, P.Z. Liang, and H. Ying, "Chinese Chess Character Recognition with Radial Harmonic Fourier Moments," *IEEE International Conference on Document and Analysis. International Conference on Document Analysis and Recognition*, 2011, pp. 1369-1373.

[5] Y.C. Wen, "Chinese-Chess Image Recognition by using Feature Comparison Techniques," *An International Journal of Applied Mathematics & Information Sciences*, 2014, pp. 2443-2453.

[6] S. Arora, D. Bhattacharjee, M. Nasipuri, D. B. Kumar, and M. Kundu, "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition," *IEEE Region 10 Colloquium and the Third ICIIS*, 2008, pp. 1-6.

[7] O. Velek, *A Likelihood Normalization Approach to Combining Multiple Classification methods and its Application to On-line Recognition of Handwritten Japanese Characters*, Department of Computer Science Tokyo University of Agriculture and Technology, 2003.

[8] O.M. Snehal and V.K. Prof. Shandilya, "Image Enhancement and its Various Techniques," *Internationa Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, 2012, pp. 370-372.

[9] H. Pujara and M.P. Kantipudi, "Image Segmentation using Learning Vector Quantization of Artificial Neural Network," *International Journal of Advanced Research in Artificial Inteligence*, vol. 2, 2013, pp. 51-55.

[10] Aaron, O.S. Sitompul, and R. F. Rahmat, "A Distributed Anonymous Neuro-Gen Learning Engine for Content-Based Document File Type Identification," *IEEE International Conference on Cyber and IT Service Management (ICITSM)*, 2014, pp. 63-68.

[11] A.S. Abdul Nasir, N. Mustafa, and N.F. Mohd Nasir, 2009. "Application of Thresholding Technique in Determining Ratio of Blood Cells for Leukimia Detection," *Proceedings of the International Conference on Man-Machine Systems (ICoMMS).*

[12] B.T. Prof. Sagar, Prof. D. Kulhare, M.D. Nirmal, and Prof. G. Prajapati, "Image Pre-processing through Erosion and Dilation Methods," *Internatonal Journal of Emerging Technology and Advanced Engineering*, vol. 3, 2013, pp. 285-289.

[13] J. Canny, "A computational approach to edge detection. Pattern Analysis and Machine Intelligence," *IEEE Transactions on PAMI*, vol. 8, 1986, pp. 679– 698.

[14] T.S. Romen, S. Roy, O.S. Imocha, T. Siman, and S. Manglem, 2011. "A New Local Adaptive Thresholding Technique in Binerization," *International Journal of Computer Science Issues*, vol. 8, 2011, pp. 271 – 277.

[15] X.Y. Liu and M. Blumentstein, A Modified Direction Feature for Cursive Character Recognition," *IEEE International Joint Conference on Neural Network (IJCNN)*, 2008.

[16] R.F. Rahmat, M. Pasha, M. Syukur, and M. Budiarto, "A Gene-Regulated Nested Neural Network," *The International Arab Journal of Information Technology*, vol. 12, 2014, pp. 531-538.

[17] N.Y. Liang, G.B. Huang, P.Saratchandran, and N.Sundararajan, "A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks," IEEE Transactions on Neural Networks, vol. 17, 2006, pp. 1411-1423.

[18] C. Erick and G.B. Huang, "Extreme Learning Machines," *IEEE Computer Society*, 2013, pp. 30-59.