

Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Network

Suthasinee Iamsa-at
Department of Computer Science
Faculty of Science, Khon Kaen University
Khon Kaen, Thailand
suthasinee@kkumail.com

Punyaphol Horata
Department of Computer Science
Faculty of Science, Khon Kaen University
Khon Kaen, Thailand
punhor1@kku.ac.th

Abstract— Feature extraction plays an essential role in hand written character recognition because of its effect on the capability of classifiers. This paper presents a framework for investigating and comparing the recognition ability of two classifiers: Deep-Learning Feedforward-Backpropagation Neural Network (DFBNN) and Extreme Learning Machine (ELM). Three data sets: Thai handwritten characters, Bangla handwritten numerals, and Devanagari handwritten numerals were studied. Each data set was divided into two categories: non-extracted and extracted features by Histograms of Oriented Gradients (HOG). The experimental results showed that using HOG to extract features can improve recognition rates of both of DFBNN and ELM. Furthermore, DFBNN provides higher slightly recognition rates than those of ELM.

Keywords—Handwritten Character Recognition; Feature extraction; Histograms of Oriented Gradients; Deep Learning Neural Network; Extreme Learning Machine

I. INTRODUCTION

Deep-Learning Neural Networks are a new type of machine learning [1] that contains the humanlike ability to learn with or without supervision. A deep-learning neural network's parameters are weight and bias. The number of parameters depends on the number of inputs and hidden nodes. G. Hinton, et al. [2] used the deep-learning neural networks for speech recognition. Y. LeCun, et al. [3] conducted an experiment on a fully-connected multilayer neural network with two layers of weight by training the Backpropagation and adjusting the weight using Stochastic Gradient Descent. They found that when the test used 300 hidden nodes, the testing errors were minimized. Another method used for character recognition is Extreme Learning Machine (ELM) [4]. Previous literature has reported that when ELM was employed to recognize handwritten Arabic Indian numerals, its classification accuracy was very high [5]. Thus, ELM was adopted into the present study to measure the classification efficiency of Thai handwritten recognition in comparison to deep learning neural networks. A previous study used feature extraction from histograms of oriented gradients (HOG) on pedestrian image detection [6]. The extraction process started by adjusting the images into grayscale, computing the gradients, weighted voting into spatial and orientation cells, and checking the

overlapping spatial blocks using L2-norm. Later, this feature extraction was then applied to handwritten characters of the Indian language Hindi [7]. Previous studies on Thai handwritten recognition efficiency had presented the hotspot technique and classified with the k-Nearest Neighbor algorithm using the Euclidean distance. It was found that the classification rate was as high as 83.3% in Thai handwritten data and 90.1% in Bangla numeric. Therefore, feature extraction from HOG was applied to extract Thai handwritten character and numeral features, especially the handwritten characters of this present study.

The present research thus focuses on investigating and comparing handwritten recognition rates, using feature extraction from HOG, between DFBNN and ELM.

II. RELATED WORK

A. Deep-Learning Feedforward-Backpropagation Neural Network (DFBNN)

Deep Learning was first presented by Geoffrey Hinton et al. [1][2] in research that used DFBNN with supervised learning, which adjusted weights using Minibatch Stochastic Gradient Descent. In a feedforward structure of an artificial neural network with more than one layer of hidden nodes each, j is the hidden node input and output layers. The logistic function performs as an activation function such as the sigmoid function,

$$y_j = \text{logistic}(x_j) = \frac{1}{1 + e^{-x_j}}. \quad (1)$$

The input y_i is sent from the previous layer to compute the output in each hidden node x_j until it reaches the output layer and y_j is computed by an activation function as in equation (1), where the input parameter of equation (1) can be computed as

$$x_j = b_j + \sum_i y_i w_{ij}, \quad (2)$$

where b_j is a bias of node j ; i is an index of every node in the previous layer, w_{ij} are weights connecting between node j in the current layer and node i in the layer below and y_j is the output of the layer below. To classify the multi-class problem, the output of the j th hidden node in the output layer transforms all the inputs of the output layer to a class probability p_j using “softmax”, which is defined as follows

$$p_j = \frac{\exp(x_j)}{\sum_{i=1}^K \exp(x_i)}, \quad (3)$$

where K is the number of classes.

Deep-learning neural networks can be trained using the backpropagation method. Upon completion of the backpropagation step, softmax via common cost function C , is used to decide the final output. The probability p_j and the target values d_j are used for computing the cost function as follows

$$C = -\sum_j d_j \log p_j, \quad (4)$$

In practice, the element of target d_j might be zero or one depending on its class. The value of C can be known as the cross entropy between the target probability d_j and the outputs softmax p_j .

A large training data set helps increase the efficiency of small derivatives calculations. This can be done by randomizing the small batch size called “minibatch” into the training before adjusting weight by Stochastic Gradient Descent. This method can be improved using “momentum” to control the direction of weight adapting. This depends on the sum between the present and previous Stochastic Gradient Descent. The principle of momentum is to increase the amount of learning, for example, the existing weight at the $t+2$ round depends on the weight at the t round and the $t+1$ round, where t is the performing round. The momentum α is a value in the range of 0 and 1, which is used for smoothing the gradient computed at the minibatch t . This approach decreases the damping oscillation and also increases the speed of convergence. In addition, η is a learning rate that is used in the equation of weights as follows

$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) - \eta \frac{\partial C}{\partial w_{ij}(t)}. \quad (5)$$

B. Extreme Learning Machine (ELM)

ELM was proposed by Guang-Bin Huang et al. [4]. ELM works on single-hidden feedforward neural networks (SLFNs).

The ELM network consists of three layers; the input layers, hidden layers and output layers. Let us assume that the hidden layer has K hidden nodes and is using an activation function $g(x)$ such as the sigmoid function. Let $\{(x_i, t_i)\}_{i=1}^N$ be N samples where $x_i = (x_{i1}, x_{i2}, \dots, x_{im})^T \in \mathbb{R}^n$. The input weights and biases are generated randomly. The ELM process will transform a non-linear equation into a linear one, as follows

$$H\beta = T. \quad (6)$$

Where $H_j = [H_j]$ ($j = 1, \dots, K$) is a vector of the output calculated from the j th hidden node, and each element of H_j is yielded from the activation function $h_{ij} = g(w_j x_i + b_j)$, A weight vector $w_{ij} = [w_{j1}, w_{j2}, \dots, w_{jn}]^T$ is a weight of the j th node and b_i is the bias of the j th hidden node. Furthermore, $\beta = [\beta_1, \beta_2, \dots, \beta_k]^T$ is the output weights in the output layer where ($j = 1, \dots, K$), $T_i = [t_1, t_2, \dots, t_m]^T$ is the target vector of the i th sample where ($i = 1, 2, \dots, N$). Therefore, the output weights can be completed as

$$\hat{\beta} = H^\dagger T, \quad (7)$$

where H^\dagger is Moore-Penrose Matrix Inverse (MP) of H and $\hat{\beta}$ is the minimum norm solution.

ELM is widely used for handwriting recognition, for example, Sabri A. Nahmoud and Sunday O. Olatunji [5] presented Arabic Indian handwritten numeral recognition and also compared the efficiency of Support Vector Machine (SVM), ELM, Hidden Markov Model (HMM) and Nearest Mean (NM). The classification efficiency rates were reported as 99.39%, 99.45%, 97.99% and 94.35%, respectively. They claimed that SVM and ELM can produce more effective recognition rates in classifying handwritten numerals than HMM and NM.

C. Histograms of oriented gradients (HOG)

HOG was proposed by Navneet Dalal and Bill Triggs [6] to present effective recognition using linear support vector machine to detect human images in a case study. The study indicated the influence of each computing step that affected the efficiency of HOG.

HOG is widely used in several applications. Aditi Goyal, Kartikay Khandelwal and Piyush Keshri [7] proposed Indian Language Hindi character-recognition by comparing the pre-processed image with the feature extracted image followed by a classifying step to design highly-effective character-recognition software. The efficiency testing accuracy rate was reported at 98.8% using the classification technique called One-against-All support vector machine and using feature extraction from HOG.

III. EXPERIMENT PREPARATION

The preparation method of the character recognition experiment of this study consists of 4 main processes: image pre-processing, feature extraction, unit vector after feature extraction, and handwritten character-recognition. A flow chart of the method is shown in Fig 1.

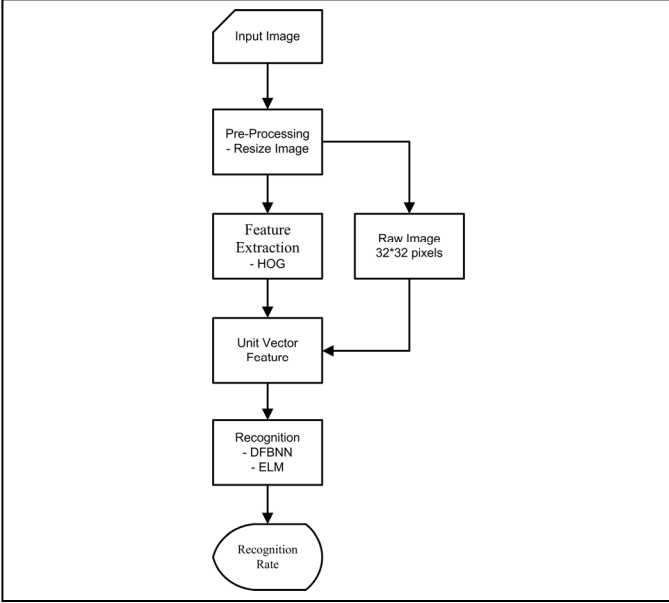


Fig. 1. A Flow Chart of Handwritten Character Recognition.

A. Data Preparation.

The present investigation used three different character sets. The first was the Thai handwritten character set called ‘Thai Characters by Olarik’ [8][9] This character set contains 743 characters and is divided into 66 classes as follows: 44 consonants, 18 vowels and 4 tone markers. The second set was the Bangla handwritten numeral set called ‘Off-Line Handwritten Bangla numerals’ [10], which contains 500 handwritten numerals and is divided into 10 classes. The third and final set was the Indian handwritten numeral set called ‘Off-Line Handwritten Devangari’ numerals [11], which contains 57 handwritten numerals and is divided into 10 classes.

B. Image Pre-Processing

The data set were all resized to a pre-determined scale. The former size was $oldX * oldY$ pixels. The images were resized to $32 * 32$ pixels. Following this, $RatioM$ and $RatioN$ were computed using the equations as follows:

$$RatioM = oldX / 32 \quad (8)$$

$$RatioN = oldY / 32 \quad (9)$$

The coordinate of the original image is replaced by $OriginalIm(i, j)$ and the resized image is replaced by $resizedIm(i, j)$. When both images were coordinated, the relationship is shown in the following equation

$$resizedIm(i,j) = OriginalIm(RatioM*i, RatioN*j) \quad (10)$$

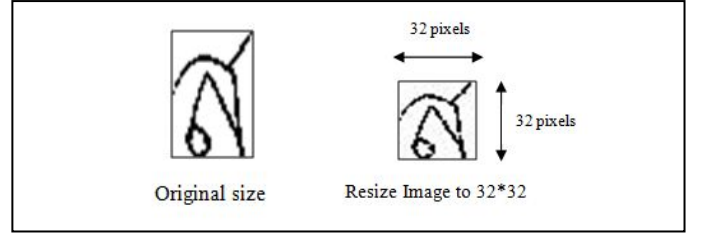


Fig. 2. The resizing of the original image to the 32x32 pixel version.

Using equation (10), an input original image is resized to the 32x32 image as shown in Fig. 2.

C. Feature Extraction

Feature extraction pulls out the dominant features of the data provided. This step is a very important step, if the representation of features are invalid then the classification might be affected. The feature extraction can be obtained using several methods. HOG is a feature extraction method that consists of the parameters shown in Table I. The parameters’ features were determined to be flexible and appropriate and to be represented without bias and variance.

The feature extraction process of HOG is started by detecting the boundary or shape of an object and normalizing them into grey scale. The gradient is computed according to the intensity of the detected object. Weighted voting is applied to the spatial and orientation cells. Lastly, the overlapping block areas are made distinct using L2 norm (normalization and descriptor blocks).

TABLE I. PARAMETERS FOR HOG FEATURES.

Parameters	Value
Filter kernels	- horizontal - vertical [-1, 0, 1] [-1, 0, 1] ^T
Number of orientation binning	- Histogram channels 9
Descriptor blocks	- Number of cells per block - Number of pixels per cell 3x3 cell blocks 6x6 pixel cells
Block normalization	- k-norm L2-norm

D. Unit Vector Features

Both of the extracted features using HOG and original features (non-extracted features) are rescaled in the range [0,1] as follows

$$Data = \left(\frac{D - \min}{\max - \min} \right), \quad (11)$$

where D is a row vector in the input features, ‘min’ and ‘max’ are the minimum and the maximum value of D, respectively.

E. Handwritten Recognition

The character features yielded from the previous process are used as the input features in the recognition process. In this process, DFBNN and ELM are employed with Thai

handwritten characters data set, Bangla handwritten numerals data set and Devanagari handwritten numerals data set.

IV. EXPERIMENTAL RESULTS

In this present study, the data sets are Thai handwritten characters containing 743 characters in 66 classes, Bangla handwritten numeral containing 500 numerals in 10 classes and Devanagari handwritten numerals containing 57 numerals in 10 classes. The training set and testing set were divided using k-Fold cross validation. The research employed 10 fold cross validation, which was simulated in the MATLAB 2011b environment running on a Intel Core i5, 2.4GHZ CPU. This method supposed that the data size equaled N samples. Each data set was divided into k parts and its size was N/k. This method used the training set for learning and data classification, which was checked using k round-training sets. For example, on the i th iteration, the i th training set was set as the testing set and the rest were training sets. Therefore, accuracy was calculated from the ratio between the whole training sets divided by the k value. For this paper, the number of hidden layers in DFBNN was set to 80 nodes and the logistic function used as an activation function. Table II. shows the parameters of DFBNN. For the training of ELM, the activation function of ELM is the sigmoid function.

TABLE II. PARAMETERS OF DEEP NEURAL NETWORK TRAINING.

Parameter	Value
Number of epochs	150
Number of batch size	4
Learning rate	0.3270
Sparsity rate	0
Sparsity target	0.05
Momentum rate	0.00015
Training time	No Limit

A. Recognition rate measurement.

The recognition rate is calculated as follows

$$\text{Accuracy} = \left(1 - \left(\frac{\text{NMS}}{\text{NS}}\right)\right) \times 100 \quad (12)$$

when

NMS is the number of miss classification samples,

NS is the number of samples.

B. Comparison Results

The results shown in Table III. show that the recognition accuracy rates of testing sets without feature extraction using DFBNN for Thai characters, Bangla numerals and Devanagari numerals datasets are 98.40%, 95.84%, and 78.40%, respectively. The recognition accuracy rates of testing sets without feature extraction from ELM for the datasets are 96.33%, 92.68%, and 65.48%, respectively. The recognition

accuracy rates of testing sets with feature extraction via HOG and trained by the DFBNN model are 99.08%, 97.40%, and 80.00%, respectively. Lastly, the recognition accuracy rates of testing sets with feature extraction by HOG of the ELM model for the datasets are 97.06%, 95.30%, and 79.60%, respectively. It can be seen that DFBNN can produce higher recognition rates than ELM for both the input extracted features and the raw input features.

TABLE III. COMPARISON OF RECOGNITION RATE OF THE NEURAL NETWORKS MODEL USING FEATURE EXTRACTION AGAINST DFBNN AND ELM WITH A RAW PIXELS AND HOG.

Data Set	Feature Extraction	Artificial Neural Network		Average Accuracy
		Model	Hidden Node	
Thai characters	Raw Pixels	DFBNN ^a	80	98.40%
		ELM ^b	500	96.33%
	HOG	DFBNN ^a	80	99.08%
		ELM ^b	500	97.06%
Bangla numerals	Raw Pixels	DFBNN ^a	80	95.84%
		ELM ^b	500	92.68%
	HOG	DFBNN ^a	80	97.40%
		ELM ^b	500	95.30%
Devanagari numerals	Raw Pixels	DFBNN ^a	80	78.40%
		ELM ^b	500	65.48%
	HOG	DFBNN ^a	80	80.00%
		ELM ^b	300	79.60%

^a Deep-Learning Feedforward-Backpropagation Neural Network. (DFBNN)

^b Extreme Learning Machine. (ELM)

TABLE IV. COMPARISON OF STANDARD DEVIATION (SD) OF THE NEURAL NETWORKS MODEL AGAINST DFBNN AND ELM.

Feature Extraction	Thai characters (SD.)		Bangla numerals (SD.)		Devanagari numerals (SD.)	
	DFBNN	ELM	DFBNN	ELM	DFBNN	ELM
Raw Pixels ^c (Image size 32*32 px)	1.0814	1.5285	0.548	3.9157	7.9179	21.1391
HOG ^d (Feature Vector Dim. = 84)	1.2932	1.5375	3.6196	4.2426	0	2.8284

^c Raw pixels without Feature Extraction

^d Feature Extraction using HOG

Table IV. shows the efficiency of the DFBNN model whose average of standard deviations of testing recognition

rates of all datasets is less than the standard deviations of ELM.

V. CONCLUSIONS

This paper presents a method for applying HOG for feature extraction and DFBNN for classification of three handwritten datasets i.e. Thai characters, Bangla numerals, and Devanagari numerals. We also studied the effects on classification in DFBNN and ELM with and without HOG. Base on the experimental results, HOG can enhance the recognition rates of both DFBNN and ELM. Furthermore, DFBNN can produce higher slightly recognition rates than ELM in all the three datasets.

REFERENCES

- [1] G. Hinton, "Learning multiple layers of representation," Trends in Cognitive Sciences, Canada, vol. 11, no.10, pp. 428 – 434, 2007.
- [2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," IEEE Signal Processing Magazine, vol. 29, pp. 82–97, Nov. 2012
- [3] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, pp. 2278-2324.
- [4] G.-B. Huang, Q.-Y. Zhu, and C.-K., "Extreme learning machine: Theory and applications", Neurocomputing, vol 70, no. 1–3, pp. 489-501, Dec. 2006.
- [5] Sabri A. Mahmoud , Sunday O. Olatunji, "Automatic Recognition of Off-line Handwritten Arabic (Indian) Numerals Using Support Vector and Extreme Learning machines," Internationnal Journal of Imaging, vol 2, no. A09, 2009
- [6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego USA, pp. 886-893, June 2005.
- [7] Aditi Goyal, Kartikay Khandelwal and Piyush Keshri. Optical Character Recognition for Handwritten Hindi. Stanford University, 2010.
- [8] O. Surinta, L. Schomaker and M. Wiering, "Handwritten character classification using the hotspot feature extraction technique," Proceedings 2012 of the 1st International Conference on Pattern Recognition Applications and Methods. Vilamoura Aalgarve Portugal, pp. 261–264, Feb. 2012
- [9] O. Surinta, Available: <https://sites.google.com/site/mrolarik/thai-character-dataset>.
- [10] U. Bhattacharya and B. B. Chaudhuri, Available: <http://www.isical.ac.in/~ujjwal/download/BanglaNumeral.html>
- [11] U. Bhattacharya and B. B. Chaudhuri, Available <http://www.isical.ac.in/~ujjwal/download/DevanagariNumeral.html>
- [12] Fungpipat P., Sunat K., Onnoom B., Wichienit N. and Chiewchanwattana S., "Performance Improvement of Scene Recognition via CENTRIST Image Descriptor and Extreme Learning Machine," 10th International Joint Conference on Computer Science and Software Engineering, Thailand, May. 2013.