Parameter Optimization of Kernel Extreme Learning Machine Using Artificial Bee Colony Algorithm and Its Application for Disease Classification

Ming-Huwi Horng[1*], Jian-Ying Cheng[1], Yu-Lun Hung[2], Yu-Cheng Hung[3], Yung-Nien Sun[4], Pongpon Nilaphruek[5]

[1] Department of Computer Science and Information Engineering, National PingTung University, Taiwan.
[2] Department of Business Computing, National Kaohsiung University of Science and Technology, Taiwan.
[3] Computer and Intelligent Robot Program for Bachelor Degree, National PingTung University, Taiwan.
[4] Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan.
[5] Computer Science Department, Rajamangala University of Technology Thanyaburi, Thailand.

* Corresponding author. Tel.:+886-87663800; email: horng@mail.nptu.edu.tw
Manuscript submitted February 24, 2020; accepted June 3, 2020.
doi: 10.17706/ijbbb.2020.10.3.127-136

Abstract: Machine learn methods have been widely used for classification and diagnosis of diseases for increasing its accuracy and efficiency. The kernel extreme learning machine is being increasingly used algorithm to training single layer forward neural network as that this network is given the weights between input and hidden layers, and the bias parameter of each hidden node. In order to obtain more stable and accurate model, an artificial bee colony algorithm is used to pre-train parameters of kernel parameter and penalty parameter. weight and bias. In this paper, an artificial bee colony based kernel extreme learning machine is proposed to classify medical datasets. This proposed method is called ABC-KELM. In experiments, we use two benchmark datasets that are Breast cancer and Parkinson disease from the UCI repository to evaluate the effectiveness and classification accuracy. The experimental results reveal that the ABC-KELM can obtain satisfactory classification results.

Key words: Machine learning, kernel extreme learning machine, artificial bee colony algorithm, UCI repository.

1. Introduction

The learning of the extreme learning machine uses the traditional gradient descent methods, but it often suffers from the slow learning speed, convergence to local optimal solutions and complicated parameters adjustment in various application [1]. Extreme learning machine (ELM) [2] is a single-hidden layer feedforward neural network with random weights between the input and the hidden layer. In general, extreme learning machine is based on empirical risk minimization principle, that is a fast learning network with remarkable generalization performance. ELM also uses the Moore-Penrose pseudoinverse for computing the weights between the hidden and the output layer which makes it fast. In some case such as limited samples, the risk minimization principal is unsatisfactory from a statistical point of view. Huang *et al*. [3] proposed an improved version called kernel extreme learning machine (KELM) to compare this model and the solution processes to original ELM. However, the KELM use the randomly generated weights between input layer and hidden layer, the bias of hidden node and learning parameters, it often requires

human intervention and could affect the classification accuracy. The learning parameters consist of the kernel parameter and the penalty parameter.

Over the last decade, modeling the behavior of social insects, such as ants and bees for searching the solution the problem solving. The artificial bee colony (ABC) algorithm is considered as a typical swarm-based approach for optimization. The ABC algorithm had been widely used in numerical optimization [4], leaf-constraint minimum spanning tree generation [5], image thresholding [6] and neural network parameter training [7]. This paper focuses in learning of kernel and penalty parameters. The parameter learning uses the artificial bee colony algorithm to decide the optimal or suboptimal parameters for disease classification. This method is called ABC-KELM method.

The remainder of this article is organized as follows. Section 2 introduces the ABC-KELM method and the material of the experiments. Experimental results and discussion are provided in Section 3. Section 4 presents conclusions and suggestions for future work.

2. Method and Materials

2.1. Kernel Extreme Learning Machine

The brief description of kernel extreme learning machine is given as follows:

Given N training samples $\{(x_i, y_i)\}_{i=1}^{N}$. The input vectors $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$, and its desired output vector $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$, T denotes the matrix/vector transpose. Here, n is the number of the input vectors that is equal to the number of input neurons, L is the number of hidden neurons, and m is the number of output neurons that is equal to the number of classes. The U matrix is the weight matrix between input and hidden layer, $U= [u_1, u_2, \dots, u_L]^T \in R^{L\times n}$ and the bias of hidden neurons, b=$[b_1, b_2, \dots, b_L]^T \in R^L$. The $u_i = [u_{i1}, u_{i2}, \dots, u_{iL}]$ are the weights connecting the ith hidden neuron, the b_i is the bias of the ith hidden neuron. The output of ith neuron of output layer o_j is modeled as

$$o_j = h(x_j) = \sum_{i=1}^{L} \beta_i h(u_i \cdot x_j + b_j), j = 1,2, \dots, m \quad (1)$$

Here, $h(\cdot)$ represents the activation function and the weight vector β_i is between the ith hidden and the output layer, o_j is the jth input data x_j target vector. $u_i \cdot x_j$ is the inner product of u_i and x_j.

Huang [3] suggested that the optimization problem of the ELM method is formulated to find the weight vectors $\beta_i, i = 1,2,..L.$, between the hidden layer and the output layer such that the $\sum_{j=1}^{N} \|o_j - t_j\|$ is minimum. The original ELM algorithm suggested the input weights, U, and the hidden layer bias, b, do not need to be tuned. Based on the assumption, the objective function can be formulated as follows:

$$\text{Minimize: } \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\|\xi_i\|^2 \quad (2)$$

Subjected to: $h(x_i)\beta = t_i^T - \xi_i^T$, *i=1,2,...,N.*

The first term of objective function is the regularization term that is also known as structure risk$\|\beta\|^2$, the second term is the least square error, which is also known the empirical risk $\|\xi_i\|^2$. The regularization parameter, C, is used to control the trade-off between the two risks.

Based on the Karush-Kuhn-Tucker (KKT) theory, the training of ELM is equivalent to solve the following dual optimization problem by minimizing the following term.

$$L_{ELM} = \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\|\xi_i\|^2 - \sum_{i=1}^{N}\alpha_i(h(x_i)\beta - t_i + \xi_i) \quad (3)$$

where each Lagrangian operator α_i corresponds to the *i*-th input vector. The optimization of Eq. (3) can be found by setting the derivatives with respect to using the following equations.

$$\beta = \sum_{i=1}^{N} \alpha_i (h(x_i)^T = H^T \alpha \tag{4}$$

$$\alpha_i \xi_i = 0, i = 1,2, \dots ., N \tag{5}$$

$$h(x_i)\beta - t_i + \xi_i = 0, i = 1,2, \dots \dots, N \tag{6}$$

where *H* is the hidden layer output matrix defined as follows:

$$H = \begin{bmatrix} h(u_1 \cdot x_1 + b_1) & \cdots & h(u_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ h(u_1 \cdot x_N + b_1) & \cdots & h(u_L \cdot x_N + b_L) \end{bmatrix} \tag{7}$$

For small training samples, the above formula can be modified as:

$$\left(\frac{I}{C} + HH^T\right)\alpha = T \tag{8}$$

where the *I* is an identity matrix, the *C* is the penalty parameter and *H* is the hidden layer output matrix.

The output weight of ELM can be computed as follows:

$$\beta = \left(\frac{I}{C} + HH^T\right)^{-1} H^T T \tag{9}$$

We can apply the Mercer's condition to efine the kernel matrix Ω_N of KELM as follows:

$$\Omega_N = H_N H_N^T : \Omega_{N_{i,j}} = h(x_i) \cdot h(x_j) = K(x_i, x_j) \tag{10}$$

where H_N is the hidden layer output matrix, H_N^T is the transpose of H_N, Ω_N is the kernel matrix, i, j ∈ {1, 2, … , N}. $K(x_i, x_i)$ are the kernel function of x_i and x_j defined as:

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2) \tag{11}$$

where γ is the parameter of kernel function.

We get the output of j-th node of output layer in the KELM model as follows:

$$o_j = \begin{bmatrix} K(x_j, x_1) \\ \vdots \\ K(x_j, x_N) \end{bmatrix}^T \left(\frac{I}{C} + \Omega_N\right)^{-1} T \tag{12}$$

2.2. Artificial Bee Colony Algorithm

The artificial bee colony (ABC) algorithm proposed by Karaboga and Basturk [8] have been become available and promising techniques for real-world optimization problems. The colony of artificial bees contains three groups of bees: employed bee, onlooker and scout. The employed bees carry the information about food source and share then in the dancing area of hive. The onlookers wait in the dancing area for

making a decision on the selection of food source depending the probability delivered by employed bees. The computation of probability is based on the amounts of the food source. The other kind of bees is scout bee that carries out random searches for new food sources. The employed bee of abandoned food becomes a scout and as soon as it finds a new food source it becomes employed bee. Therefore, each cycle of the ABC algorithm contains three steps. First, the employed bees are sent into the food source and the amounts of nectar are calculated. After sharing the information of food sources, the onlooker bees visit food sources and then updates them. A scout is sent out to find new food source when the depletion of nectar of a food source occurs.

In the algorithm, the position of a food source x_i represents a candidate solution of the optimization problems and their amount of nectar in this food source is denoted as the fitness $fit(x_i)$. In general, the number of employed bees or onlookers is equal to the number of the food sources. Initially, the ABC algorithm randomly generates a distributed initial population $P = \{x_1, x_2, \dots, x_K\}$ of K solutions, where K also denoted the number of employed bees or onlookers. Each solution x_i *(i=1, 2,..., K)* is a D-dimensional vector. In each execution cycle, C (C=1, 2,..., MCN), the population of the solutions is subjected to the search processes of the employed bees, onlookers and scouts. An employed bee modifies the possible solution depending on the amount of nectar (fitness value) of the new solution (food source) by using the Eq. (13).

$$v_{ij} = x_{ij} + \varphi(x_{ij} - x_{kj}) \tag{13}$$

where $k \in \{1,2, \dots, K\}$, but $k \neq I$ and $j \in \{1, 2, \dots, D\}$ are randomly selected index, φ is a random number between [-1, 1].

If the nectar the new solution v_i is more than the previous solution x_i, the employed remembers the new solution and synchronously abandons the old one, otherwise it remains the location of previous one in its mind.

When all employed bees have finished their search process, they bring back the nectar information and position of all food sources to the onlookers, each of them decides a food source for further calling upon according to a probability proportional to the amount of nectar in that food source. The probability p_i of selecting a food source z_i is determined using the following Eq. (14).

$$p_i = \frac{fit(z_i)}{\sum_{n=1}^{K} fit(z_n)} \tag{14}$$

In practice, each food source, z_i, sequential generates a random number between [0, 1], and if the random number is less than the probability p_i, an onlooker is sent to the food source and produces a new solution based on the Eq. (15).

$$v_{ij} = z_{ij} + \varphi(z_{ij} - z_{kj}) \tag{15}$$

where $k \in \{1,2, \dots, K\}$, but $k \neq i$ and $j \in \{1, 2, \dots, D\}$ are randomly selected index, φ is a random number between [-1, 1].

If the fitness of the new solution is more than the old one, the onlooker memories the new solution and shares this information with other onlookers. Otherwise, the new solution will be discarded. The process is repeated until all onlookers have been distributed to food sources.

If the food source could be improved through the predetermined number “limit”, then the food source is abandoned and then the corresponding employed bee become a scout. The scout discovers a new food source to replace with the abandoned solution z_j by using the Eq. (16).

$$z_{ij} = z_{min}^{j} + \sigma\left(z_{max}^{j} - z_{min}^{j}\right) \quad (16)$$

where z_{min}^{j} and z_{max}^{j} are the lower and upper bound of the *j*-th component of solutions. The σ is a random number ranging with -1 and 1. If the new solution is better than the abandoned solution z_j, the scout becomes employed bee and the new solution is generated.

The search of employed bee, onlooker and scout is repeated until the execution cycle equals to MCN. So far, the best solution with largest fitness are outputted as the solution of the ABC algorithm.

3. Artificial Bee Colony Based Kernel Extreme Learning Machine

In this section, we introduce the training of the kernel extreme learning machine is essentially a constrained optimization problem. The constrained optimization must first decides the fitness function of each solution z_i. It is clear that the ABC-KELM algorithm starts with a set of initial population (candidate solutions). The solution z_i is simulated as a vector comprising optimization parameters of the kernel parameter and penalty parameter. The populations of *m* initial solutions are randomly generated with 2 dimensional denoted by *P*:

$$\begin{aligned} &\mathrm{P}=\{z_1, z, \ldots, z_m\} \\ &z_i = (\gamma^i, C^i), \end{aligned} \quad (17)$$

where $\gamma^i, C^i \in R$, γ^i and C^i are the smoothness parameter of the kernel function and penalty parameter of solution z_i, respectively.

The output and desired output of any one training sample x_i are o_j and t_i, respectively. The square error of sample x_i us defined as $e_i = (t_i - o_i)^2$. The mean square error (MSE) of training set in a training set with *l* size can be defined as the Eq. (18).

$$\mathrm{MSE}=\frac{1}{l}\sum_{i=1}^{l}(t_i - o_i)^2 \quad (18)$$

The fitness of training set can be written as Eq. (19).

$$\mathrm{Fit}(z_i) = \frac{1}{1+MSE} \quad (19)$$

The details of the proposed algorithm are described as follows.

- *Step 1. Generate the initial population of solutions of ABC-OCSVM algorithm.*

Generating the m solutions z_i (*i*=1, 2…, m) with 2 dimensions denoted by matrix P defined as the Eq. (17). The fitness of each solution is evaluated by Eq. (19) and then set cycle =1 and the trial number of each solution, $trail_i$, is equal to 0.

- *Step 2. Place the employed bees to their food sources.*

In step 2, each of employed bees is sent to one of solutions, z_i, then produces a new solution v_i by using the Eq. (13) and computes the fitness of the new solution. If the fitness of the new one is larger than that of the previous old one, the employed bee memorizes the new solution and updates the old one; otherwise, the employed bee keeps the old solution.

- *Step 3. Send the onlooker to their food solution.*

In step 3, we first calculated the probability value p_i of the solution z_i according the solutions delivered by the employed bees using Eq. (14). An onlooker bee decides a solution to update it depending on the probabilities and then determines a neighbor solution around the chosen one. In the solution selection of onlookers, each onlooker randomly generates a randomly number ranged from 0 to 1. If the random

number is less than p_i, the ABC-KELM algorithm sends the onlookers to these undelivered solutions and updates them according to the Eq. (15) just as the employed bees do.

- *Step 4. Send the scout to search for new when a solution is abandoned.*

If the solution z_i is not improved through steps 2 and 3, the $trail_i$ is increased by one. If the $trail_i$ is more than the predetermined parameter "limit", the z_i is considered to be an abandoned solution. The employed bee of this solution is changed into a scout to search for a new solution. The scout first randomly produces the new solution by Eq. (16). If the fitness of new solution is more than the abandoned solution, the scout becomes employed bee and the new solution is generated.

- *Step 5. Recorded the best solution.*

In this step, the best solution so far is recorded and increases the cycle by 1.

- *Step 6. Check the termination condition.*

If the cycle equals to the maximum cycle number (MCN) then the algorithm is finished and outputs the best solution; otherwise go to Step 2.

4. Experimental Results and Discussion

We implement all of the algorithm in Python 3.0 version on a personal computer with 2.4 GHz, 16G RAM running window 10 system. Three control parameters that are the number of food sources which is equal to the number of employed or onlookers, the value of "limit" and the maximum cycle number (MCN) are set in n=30, MCN 100, and "limit" 10.

Two medical databases that are the Parkinsons and breast cancers datasets of UCI data repository [9] are used in the experiments. The description of the two datasets is listed in Table 1. The features of data of the two datasets are real number and they are normalized into 0 to 1 for later training. The normalization makes the distribution of feature consistent. All experiments are conducted by the five-fold cross-validation method, which was performed using the exhaustive search with a leave-one-out approach. Three performance indices that are accuracy, sensitivity and specificity are used to compare to work of the LIBSVM 3.2.2 version proposed by Lin [10]. The three indices are defined as below:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{20}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{21}$$

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{22}$$

TP, TN, FP, and FN denote the number of true positive, true negative, false positive and false negative, respectively.

Table 1. The Parkinsons and Breast Cancer Datasets

Dataset	Attribution Dimensions	Instance Number	Positive Number	Negative Number
Parkinsons	23	195	48	147
Breast cancer	10	699	458	241

4.1. Experiments of Breast Cancer Dataset

Fig. 1 shows the diagram of mean square error between the desired output and the true output with respect to the iterative number. It is evident that the ABC-KELM algorithm can converge as the iterative number reaches 35. Table 2 shows the learned results of kernel parameter (γ), penalty parameters (C) and execution times of the first one fold cross-validation.

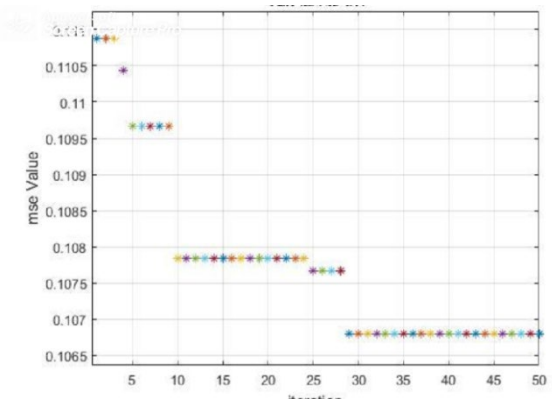


Fig. 1. The relation between the error measure vs. iteration.

Table 2. The Trained Parameters in the First One Fold of Dataset

Kernel Parameter	Penalty Parameter	Execution Time
1.9262	2.2955	3.23 (seconds)

Table 3. The Classification Results of the Five Folds in the Cross-Validation Method

Fold Number	TP	TN	FP	FN
1	90	47	1	1
2	90	46	1	2
3	90	46	1	2
4	89	47	2	1
5	88	47	3	1
Total	447	233	8	7

In Table 3, we find the total number of false positive and false negative are 8 and 7. The proposed ABC-KLEM achieved high performance measurements in the breast cancer dataset, with an average of accuracy, specificity and sensitivity of 0.938, 0.970 and 0.9835, respectively.

4.2. Experimental Results of the Parkinsons Dataset

Fig. 2 shows the diagram of mean square error between the desired output and the true output with respect to the iterative number. It is evident that the ABC-KELM algorithm can converge as the iterative number reaches 15. Table 4 shows the learned results of kernel parameter (γ), penalty parameters (C) and execution times of the first one fold cross-validation.

Table 4. The Trained Parameters in the First One Fold of Dataset

Kernel Parameter	Penalty Parameter	Execution Time
3.9262	1.2955	2.98 (seconds)

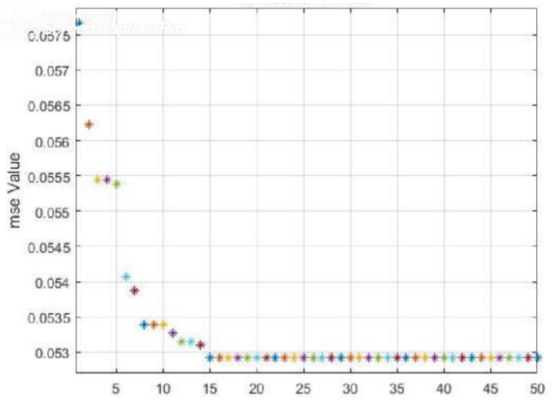


Fig. 2. The relation between the error measure vs. iteration.

Table 5. The Classification Results of the Five Folds in the Cross-Validation Method

Fold Number	TP	TN	FP	FN
1	8	28	1	1
2	9	25	0	4
3	9	26	0	3
4	8	26	1	3
5	7	27	2	2
Total	41	132	4	13

In Table 5, we find the total number of false positive and false negative are 3 and 13. The proposed ABC-KLEM achieved high performance measurements in the breast cancer dataset, with an average of accuracy, specificity and sensitivity of 0.915, 0.971 and 0.760, respectively. Obviously, the sensitivity is only 0.76. In the Parkinsons dataset, the positive data is only a quarter to the negative one. It may suffer from the problem of sample imbalance.

5. Conclusion

In this paper, ABC-KELM is proposed for classification of two benchmark medical datasets. The ABC-KEML pays attention to the learning of two parameters: kernel parameter and penalty parameter. Experimental results show that the usage of ABC-KELM method can reaches accuracy 0.938 for breast cancer dataset and 0.915 for Parkinsons dataset.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The Prof. Ming-Huwi Horng and Mr. Pongpon Nilaphruek wrote the paper, Mr. Jian-Ying Cheng, Mr. Yu-Lun Hung and Ms. Yu-Cheng Hung conducted the all programs and research, Prof. Yung Nien Sun analyzed the data. Finally, all authors had approved the final version.

Acknowledgment

The authors thank the Ministry of Science and Technology, ROC (Project number: MOST-108-2634-F-006-005 and MOST-107-2622-8-006-015) and the National PingTung University (Project number: NPTU-108-003) for support of this work.

References

[1] He, Z., Wem, X., Liu, H., & Du, J. (2004). A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region. *J. Hydrol.*, *509*, 279-386.

[2] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neuro-computing*, *70(1-3)*, 489-501.

[3] Huang, G., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern Part B*, *42(2)*, 512-529.

[4] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony algorithm. *Journal of Global Optimization*, *39*, 459-471.

[5] Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, *9(2)*, 625-631.

[6] Horng, M. H. (2011). Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. *Expert Systems with Applications*, *38*, 13785-13791.

[7] Karaboga, D., & Bahriye, A. (2007). Artificial bee colony algorithm on training artificial neural network. *Proceedings of IEEE 15th Signal Processing and Communication Application Conference*, 11-13.

[8] Karaboga, D., & Akay, B. (2009). A survey: Algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, *31(1)*, 68-85.

[9] Dua, D., & Graff, C. (2019). UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science.

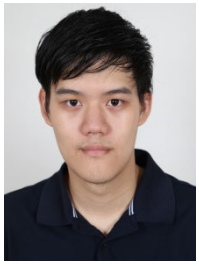
Yung-Nien Sun received his BS degree from National Chiao-Tung University, Hsin-chu, Taiwan, in 1978, and his MS and PhD degrees from University of Pittsburgh, Pittsburgh, PA, in 1983 and 1987, respectively. He was an assistant scientist with the Brookhaven National Laboratory, NY, from 1987 to 1989. Since 1989, he joined the Faculty of the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, as an associate professor, became a professor in 1993, and is currently a distinguished professor with both the Department of Computer Science and Information Engineering and the Institute of Medical Informatics. He has been working on medical image/signal analysis and industrial vision since 1982, and has published more than 200 paper, approximately half of them in referred journals. Sun is a senior member of IEEE, member of Sigma Xi, the Chinese Association of Image Processing and Pattern Recognition, and the Chinese Association of Biomedical Engineering. Yung-Nien Sun is currently the director of MOST AI Biomedical Research Center in NCKU, Taiwan.

Ming-Huwi Horng received his BS degree from National Cheng-Kung University, Taiwan, in 1990, and his MS and PhD degrees from National Cheng-Kung University, Taiwan, in 1992 and 1997, respectively. Currently, he is a distinguished professor with with Department of Computer Science and Information Engineering, National Pingtung University. His main researches include medical image/signal analysis, deep learning, machine learning and artificial intelligence.

Pongpon Nilaphruek received his MS degree from Asian Institute of Technology, Bangkok, Thailand. Currently, he joins the Department of Mathematics and Computer Science and Technology, Rajamangala University of Technology Thanyaburi, Thailand. His main research area includes software development, application on smart devices.

Jian-Ying Cheng received his BS degree from Tajen University and MS degree from Department of Computer Science and Information Engineering, National Pingtung University. His main researches include image/signal analysis, machine learning and artificial intelligence.

Yu-Cheng Hung pursuits her BS degree from the Computer and Intelligent Robot Program for bachelor degree, National PingTung University, Taiwan. Her main researches are include robot program, machine learning and artificial intelligence.



Yu-Lun Hung pursuits his BS degree from the Department of Business Computing, National Kaohsiung University of Science and Technology, Taiwan. His main researches are include image processing, machine learning, artificial intelligence and the business computing.