Development of a Nutrient Deficiency Detection System for Rice Plants Using a Co-Evolutionary PSO Algorithm

Riza Alfita[1, a)], Achmad F. Ibadillah[1,b)], Rosida Vivin Nahari[1,c)]

Author Affiliations
[1]*Faculty of Engineering, Universitas Trunojoyo, Bangkalan, Indonesia*
[a)]Corresponding author: riza.alfita@trunojoyo.ac.id
[b)]fiqhi.ibadillah@trunojoyo.ac.id
[c)]rosida.nahari@trunojoyo.ac.id

Abstract. Rice plants that receive balanced nutrition can significantly increase crop success and food production. Deep learning technology has the potential to identify nutrient deficiencies in plants by analyzing rice leaf images. This study examines the weaknesses of three rice nutrients, namely Nitrogen, Phosphorus, and Potassium using a CNN model. However, conventional CNN-based systems for detecting these deficiencies face challenges in determining optimal parameter values before training. The choice of the right model parameters greatly affects model performance. This study uses a co-evolution approach, combining Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), to optimize the parameters of the Convolutional Neural Network (CNN) used as a detection model. The co-evolution process allows the CNN model and PSO-GA parameters to evolve simultaneously, leading to a more accurate and efficient model. Experimental results show that the developed system can classify healthy and nutrient-deficient rice plants with an impressive accuracy of 95.4%, surpassing the previous accuracy of 87.7%.

INTRODUCTION

Rice, as the main source of carbohydrates for most of the world's population, is highly dependent on the availability of sufficient nutrients for its growth and development. One of the factors that cause nutritional deficiencies in rice includes a lack of macronutrients (nitrogen, phosphorus, potassium) [1]. The direct impact of nutritional deficiencies in rice plants for farmers is a decrease in crop yields which also has an impact on reducing community food production. So it is necessary to build information technology to identify nutrient deficiencies in rice plants. [2]

Not all farmers have sufficient knowledge and technical skills to operate modern equipment and interpret analysis results. Meanwhile, testing soil and plant samples in the laboratory also tends to be expensive. The solution to this problem is to use deep learning-based information technology, namely CNN, to detect nutritional deficiencies in rice. [3,4]

The best performance of the CNN method is influenced by the configuration of external variables or CNN training parameters [5], Parameters that cannot be changed during the training process of a CNN model are called Hyperparameters. [6-8] To be able to determine hyperparameters, this can be done using manual tuning before the CNN training process. The PSO method is generally used for hyperparameter optimization. [9], The weakness of this algorithm is that it tends to get stuck in local optimal solutions, namely the best solution found in a certain search area, but not necessarily the best solution globally. GA is an optimization method inspired by the process of biological evolution. The advantage of GA is that GA can perform global searches, so it can find better optimal solutions. [10]

This research uses a combination of PSO and GA methods to optimize CNN hyperparameters so that it can produce optimal accuracy in identifying rice nutritional deficiencies. [11].

MATERIAL AND METHOD

The research method used in this research starts from using a dataset of rice nutritional deficiencies obtained via the Kaggle site. Next, carry out pre-processing on the rice image dataset [12]. Then proceed with data training using CNN while simultaneously optimizing hyperparameters with the proposed method, namely co-evolutionary PSO

[13,14]. The final stage is to evaluate the CNN model using a confusion matrix to determine the level of accuracy, precision, recall, and f1-score. The flow of research methods can be seen in Figure 1 [15].

The research method used in this research starts from using a dataset of rice nutritional deficiencies obtained via the Kaggle site. Next, carry out pre-processing on the rice image dataset [12]. Then proceed with data training using CNN while simultaneously optimizing hyperparameters with the proposed method, namely co-evolutionary PSO [13,14]. The final stage is to evaluate the CNN model using a confusion matrix to determine the level of accuracy, precision, recall, and f1-score. The flow of research methods can be seen in Figure 1. [15].

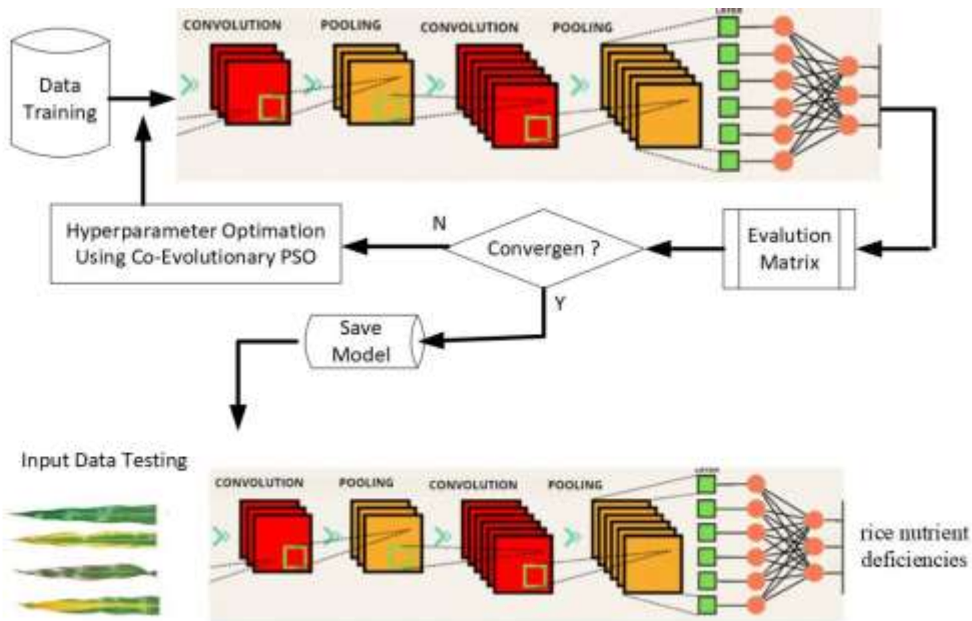

FIGURE 1. Research Flow

Image Dataset

The dataset used in this research was taken from Kaggle with a total of 1156 images of rice leaves.

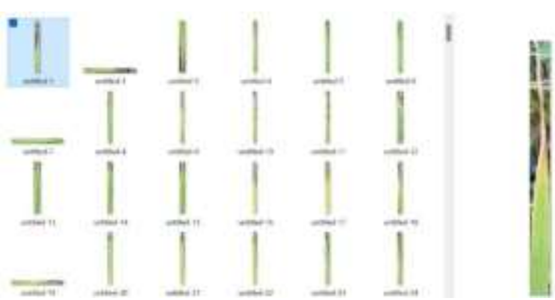

FIGURE 2. Nitrogen Deficiency of Rice

FIGURE 3. Potassium Deficiency of Rice

FIGURE 4. Phosphorus Deficiency of Rice

Convolutional Neural Network (CNN) Architecture

Generally, CNN has three main types of layers, namely Convolutional Layer, Pooling Layer, and Fully Connected Layer. (16) Convolutional layers are the core building blocks of CNN which aim to learn feature representation from input where this layer consists of several kernels or filters. Used to calculate feature maps. A new feature map is generated by combining the input with a kernel and applying activation functions such as Sigmoid, ReLu, etc. to the convolution results (17), (18). The pooling layer takes a small region of the convolution output as input and continues it to produce a single output. There are various pooling layer techniques like max-pooling, minpooling, average pooling, etc. Pooling layers reduce the number of parameters to be calculated and make the network translation invariant. The final part of a CNN consists of one or more Fully Connected Layers. A fully Connected Layer takes input from the pooling layer and produces output. (19), (20).

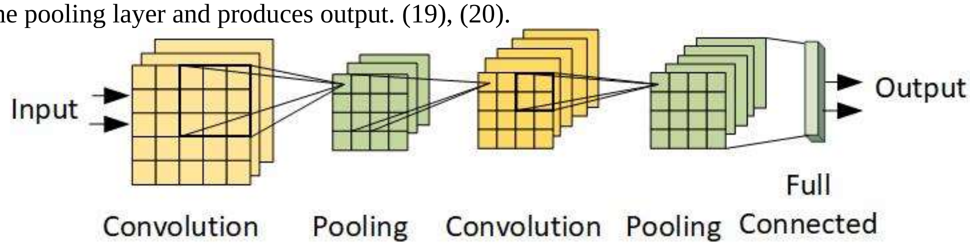

FIGURE 5. CNN Architecture

Hyperparameter CNN

Hyperparameters are parameters whose values are not learned from the data. CNN has main hyperparameters such as learning rate, batch size, and epoch. To get the best performance, it is necessary to perform hyperparameter tuning. Next, we will explain the optimization method for hyperparameter tuning which uses the Particle Swarm Optimization (PSO) algorithm and combines it with the Genetic Algorithm (GA). (20)

Particle Swarm Optimization (PSO)

Particle swarm optimization is a branch of evolutionary algorithms. PSO is based on the behavior of a flock of birds or fish. When a herd does not have a leader to find food they will scatter randomly to find where the food is. This algorithm is based on the social behavior of this organism. Social behavior consists of individual actions and the influence of other individuals (21).

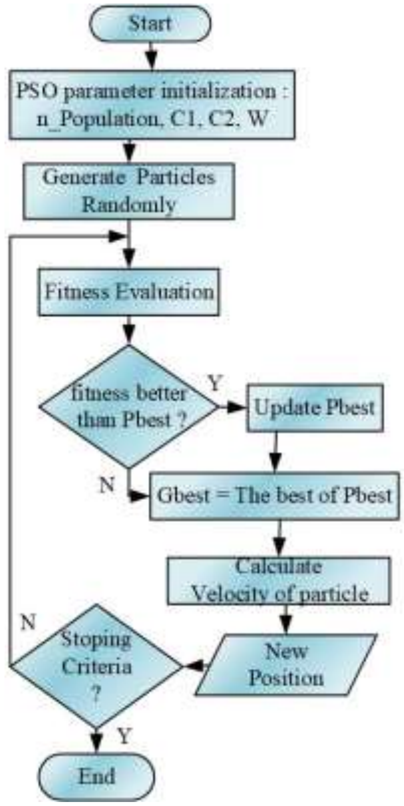

FIGURE 6. Particle Swarm Optimization (PSO) Algorithm

In this PSO algorithm, the search for solutions is carried out by a population consisting of several particles. The population is generated randomly with the limits of the smallest value and the largest value. Each particle represents the position and location of the problem at hand. Each particle searches for the optimal solution using the intelligence of the individual's experience by traversing the dimensions of the search space D. This is done by each particle making adjustments to the best position of the particle (local best) and adjusting the position of the best particle from the best value of all flock (global best) while traversing the search space. At each iteration, its performance is evaluated for each solution represented by the particle position by inserting the solution into a fitness function. Each particle is treated like a point in a certain space dimension. Then there are two factors that characterize the status of the particle in the search space, namely the X position and the Y velocity of the particle.

The following is an equation that describes position and velocity:

$$X_i(t) = x_{i1}(t), x_{i2}(t), x_{i3}(t), \dots, x_{iN}(t)$$
$$V_i(t) = v_{i1}(t), v_{i2}(t), v_{i3}(t), \dots, v_{iN}(t)$$

Where X is the position of the particle. V is the particle velocity. i and t are the particle index and the t-th iteration, in N space dimensions. The following is a mathematical model that describes the mechanism for improving particle status.

$$V_i(t) = wV_i(t-1) + c_1 r_1 \left(X_i^L X_i(t-1) \right) + c_2 r_2 (X^G - X_i(t-1))$$
$$X_i(t) = V_i(t) + X_i(t-1)$$

$X_i^L = X_{i1}^L, X_{i2}^L, ..., X_{iN}^L$ represents the local best of the ith particle. Meanwhile, $X^G = X_1^G, X_2^G, ..., X_N^G$ represents the global best of the entire flock. c1 and c2 are constants with positive values which are usually called learning factors. r1 and r2 are positive random numbers with values between 0 and 1. w is the inertial parameter. The fourth equationis used to obtain a new particle velocity based on the previous velocity, the distance between the current position and the particle's best position (local best), and the current distance to the best position of the flock (global best). Then the particles fly to the new position based on Equation 2. The workflow of PSO can be seen in Fig. 6.

Genetic Algorithm (GA)

Genetic Algorithm (GA) is a genetic algorithm included in the group of evolutionary algorithms. This algorithm was first introduced by Holland in 1975 and is a commonly used search method and was inspired by population genetics in finding a solution to a problem. This algorithm also follows Charles Darwin's concept with his theory of evolution where strong individuals will survive in the population. The basic elements of natural genetics are natural selection, crossbreeding, and mutation. (22), (23)

Natural selection is an attempt to maintain the best individuals by multiplying the best individuals. So that the best individuals are not lost in subsequent iterations. Crossbreeding operators are used to create new individuals. To create a new individual, two parents are needed. The most frequently used master selection technique is the roulette wheel. The mutation operator is used to replace the worst individual with a new individual. The number of individuals replaced depends on the mutation ratio parameter.

Co-Evolutionary PSO Algorithm

Co-Evolutionary Particle Swarm Optimization (Co-PSO) is a hybrid optimization algorithm that combines the principles of Particle Swarm Optimization (PSO) with co-evolutionary concepts. This approach aims to enhance the exploration and exploitation capabilities of traditional PSO by introducing multiple, interacting populations of particles.

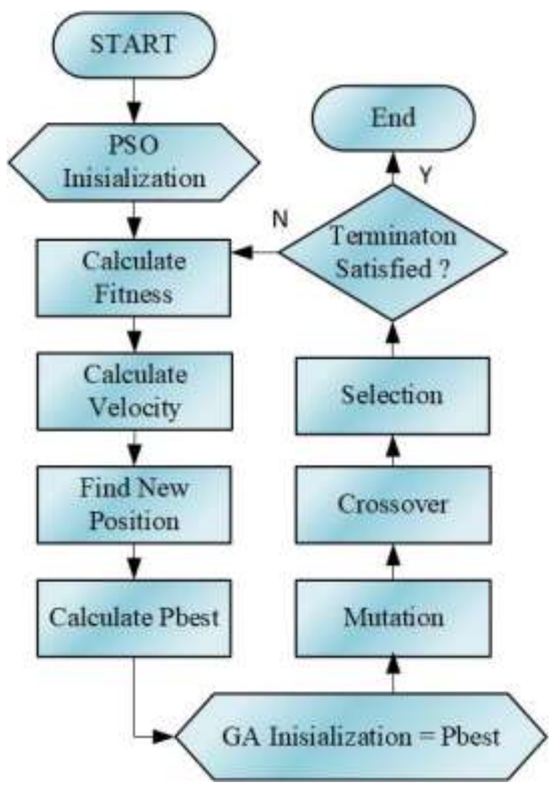

FIGURE 7 . Co-Evolutionary PDO Algorithm

1. Initialize PSO parameters.
2. Calculate Fitness value.
3. Calculate velocity.
4. With the new velocity obtained, update the position of each individual
5. Determine the Pbest value
6. Pbest from the PSO algorithm is used as the individual initialization in the Genetic Algorithm (GA)
7. GA Mutation process and continued with Crossover for candidate pairs
8. Selection of crossover pairs
9. The process is repeated until the stopping criteria are met.

Model Evaluation

This stage aims to evaluate the performance of the CNN model. Model evaluation will be carried out using a confusion matrix to determine the accuracy, precision, recall and f1-score values of the model used. The table of confusion matrix to measure CNN performance is shown on table 2.

Table 2. Confussion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Postive (1)	True Positive (TP)	False Positive (FP)
Predicted Negative (0)	False Negative (FN)	True Negative (TN)

The evaluation carried out in this research was calculated using 3 measurement matrices, namely:

1. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
2. $Precision = \frac{TP}{TP+FP}$
3. $Sensitivity\ or\ Recall = \frac{TP}{TP+FN}$

RESULTS AND DISCUSSION

This section explains the results of the experiments that have been carried out. The experiments were carried out by training the training data and validation data using the CNN architecture. In the training stage, the CNN hyperparameters were optimized first to obtain a CNN model with better performance. This model will be compared with a CNN model that does not use hyperparameter optimization.

The following is a table for the initial initialization of the CNN hyperparameters that will be tested.

Table.3 Initialization of Candidate Hyperparameter Pairs

Hyperparameter	Candidate Solution
Learning Rate	[0.001, 0.01, 0.1]
Batch Size	[10,15,25]
Epoch	[16,32,64]

Scenario 1

The trial was conducted to see the effect of using candidate pairs of hyperparameters learning rate, batch size and epoch through training using a simple neural network architecture. The maximum iteration for the Co-Evolutionary PSO optimizer is 30 iterations.

Table 4. Experiment Results of Scenario 1

Hyperparameter Candidate Pairs						Accuracy
Range	**Learning Rate**	**Range**	**Batch Size**	**Range**	**epoch**	
lb=0,001, ub=0,01	0,00543	lb=10, ub=15	12,68718	lb=16, ub=32	29,96805	0,88009
	0,00155		10,48472		22,71477	0,89497
	0,00733		12,42202		23,82776	0,89229
	0,00767		14,03856		19,85035	0,90414
	0,00872		11,98551		30,19875	0,90281
	0,00432		10,33840		17,88169	0,89203
	0,00390		12,64173		28,39220	0,91142
	0,00265		14,11433		21,72125	0,86440
	0,00608		12,74921		26,55601	0,89122
	0,00137		10,60309		30,16991	0,91545
	0,00638		12,72168		17,81105	0,90255
	0,00657		12,09399		31,23688	0,90527
	0,00879		13,03515		30,28925	0,85964
	0,00171		13,27174		31,44566	0,89374
	0,00376		10,83158		30,43223	0,90486
lb=0,01, ub=0,1	0,05117	lb=15, ub=25	15,49252	lb=32, ub=64	36,82539	0,90545
	0,06933		24,46883		37,00900	0,85750
	0,01844		15,79583		35,48345	0,97987
	0,03495		20,08661		36,13770	0,89048
	0,06811		21,35178		54,76567	0,89589
	0,01576		24,37986		60,31278	0,90522
	0,05016		22,43540		55,35141	0,90602
	0,05916		22,21097		51,79943	0,88524
	0,06819		24,89962		60,00065	0,86836
	0,08959		19,48560		49,41710	0,90938
	0,09643		15,55460		60,20231	0,85290
	0,08093		23,15417		52,89655	0,90888
	0,08839		17,54665		39,53724	0,86027
	0,06104		22,08983		55,62713	0,86467
	0,08005		19,17328		50,68414	0,89895

Based on the trial results table in scenario 1, the best candidate pair of hyperparameters resulting from the Co-Evolutionary PSO algorithm is using a learning rate of 0.01844, Batch size 15 and 35 epochs with an accuracy of 97.9%.

Scenario 2

The trial was conducted by comparing the accuracy value of the proposed model with the accuracy value of the conventional CNN model without using the hyperparameter optimizer. In this conventional CNN, using the Hyperparameter learning rate value of 0.001, Batch size 10 and 30 epochs. After testing using the conventional CNN model to detect rice nutrient deficiencies, an accuracy of 87.7% was obtained

.

Tabel 5. Confussion Matrix CNN Konvensional

	Positive (1)	Negative (0)
Positive (1)	899	61
Negative (0)	81	115

The CNN Co-Evolutionary PSO model detects rice nutrition using the best hyperparameters from the trial results in scenario 1 resulting in an accuracy of 95.4%

Table 6. Confusion Matrix CNN Co-Evolutionary PSO

	Positive (1)	Negative (0)
Positive (1)	1022	21
Negative (0)	32	81

CONCLUSION

This Research optimizes the CNN model using several hyperparameters, namely the number of Convolutional layers, kernel size and the number of filters in each layer for detecting rice nutrient deficiencies. This study aims to obtain optimal hyperparameters so as to provide good performance on the CNN model. Based on the experiments that have been carried out, the determination of hyperparameters greatly affects model performance. Hyperparameters with a combination of Learning rate 0,01844, Batch Size 15, and epoch 35 provide the most optimal results, with an accuracy value of 95,4%, precision of 97,9%, recall of 96,9%, and f1-score 97,4% on the testing data used. This shows that the model can classify the testing data according to class.

ACKNOWLEDGMENTS.

The research team would like to thank Universitas Trunojoyo Madura for giving us the opportunity to take part in science development activities in the field of computer vision

REFERENCES

1. Aulia DOP, Rizal S, Kumalasari NCP. Klasifikasi Gejala Defisiensi Nutrisi Pada Tanaman Padi Menggunakan CNN Dengan Arsitektur Resnet-50. Univ Telkom. 2022;8.
2. Alindi DY, Idmayanti R, Lestari T. Penerapan Sistem Pakar Diagnosa Penyakit Pada Tanaman Cabai Menggunakan Metode Forward Chaining Berbasis Android. JITSI J Ilm Teknol Sist Inf. 2023;4(2).
3. Yousef M, Allmer J. Deep learning in bioinformatics. Turkish J Biol. 2023;47(6).
4. Janiesch C, Zschech P, Heinrich K. Machine learning and deep learning. Electron Mark. 2021;31(3).
5. Xu T, Yuan Z. Convolution Neural Network with Coordinate Attention for the Automatic Detection of Pulmonary Tuberculosis Images on Chest X-Rays. IEEE Access. 2022;10.
6. Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. Vol. 13, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2023.
7. Saad MH, Hashima S, Sayed W, El-Shazly EH, Madian AH, Fouda MM. Early Diagnosis of COVID-19 Images Using Optimal CNN Hyperparameters. Diagnostics. 2023;13(1).
8. MINARNO AE, MANDIRI MHC, ALFARIZY MR. Klasifikasi COVID-19 menggunakan Filter Gabor dan

CNN dengan Hyperparameter Tuning. ELKOMIKA J Tek Energi Elektr Tek Telekomun Tek Elektron. 2021;9(3).
9. Jain M, Saihjpal V, Singh N, Singh SB. An Overview of Variants and Advancements of PSO Algorithm. Vol. 12, Applied Sciences (Switzerland). 2022.
10. Sakthipriya S, Naresh R. Precision agriculture based on convolutional neural network in rice production nutrient management using machine learning genetic algorithm. Eng Appl Artif Intell. 2024;130.
11. Kilichev D, Kim W. Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO. Mathematics. 2023;11(17).
12. Hong S, Jiang Z, Liu L, Wang J, Zhou L, Xu J. Improved Mask R-CNN Combined with Otsu Preprocessing for Rice Panicle Detection and Segmentation. Appl Sci. 2022;12(22).
13. Tian Y, Gao S, Li H, Dai Q, Yang J. Particle swarm intelligence promotes cooperation by adapting interaction radii in co-evolutionary games. Appl Math Comput. 2024;474.
14. Zhang Y, Zhou M, Luo W, Cheng Z. An Adaptive Learning Co-Evolutionary Variational Particle Swarm Optimization Algorithm for Parameter Identification of PMSWG. Prog Electromagn Res C. 2024;141.
15. Komal, Sethi GK, Bawa RK. A Hybrid Approach of Preprocessing and Segmentation Techniques in Automatic Rice Variety Identification System. J Sci Res. 2022;14(1).
16. Rahman ZU, Ullah SI, Salam A, Rahman T, Khan I, Niazi B. Automated Detection of Rehabilitation Exercise by Stroke Patients Using 3-Layer CNN-LSTM Model. Vol. 2022, Journal of Healthcare Engineering. 2022.
17. Ren S, He K, Girshick R, Zhang X, Sun J. Object detection networks on convolutional feature maps. IEEE Trans Pattern Anal Mach Intell. 2017;39(7).
18. Azam S, Montaha S, Fahim KU, Rafid AKMRH, Mukta MSH, Jonkman M. Using feature maps to unpack the CNN ‘Black box’ theory with two medical datasets of different modality. Intell Syst with Appl. 2023;18.
19. Kalaycı TA, Asan U. Improving Classification Performance of Fully Connected Layers by Fuzzy Clustering in Transformed Feature Space. Symmetry (Basel). 2022;14(4).
20. Xu Q, Zhang M, Gu Z, Pan G. Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs. Neurocomputing. 2019;328.
21. Dharmawan WS. KOMPARASI ALGORITMA KLASIFIKASI SVM-PSO DAN C4.5-PSO DALAM PREDIKSI PENYAKIT JANTUNG. I N F O R M A T I K A. 2022;13(2).
22. Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: past, present, and future. Multimed Tools Appl. 2021;80(5).
23. Khan A, Khan A, Ullah M, Alam MM, Bangash JI, Suud MM. A computational classification method of breast cancer images using the VGGNet model. Front Comput Neurosci. 2022;16.