Pak. J. Bot., 58(4) April, 2026.

ARTICLE DOI: http://dx.doi.org/10.30848/PJB2026-4(17)

APPLICATION OF NNE-CNN FOR NON-NUMERICAL INDIVIDUAL FEATURE EXTRACTION IN RICE BREEDING OPTIMIZATION

XIAO HAN[1], QINGRUI ZHANG[1], ZITING GAO[1], XIAOLIANG HE[2] AND FENGLOU LING[1*]

[1]College of Agriculture, Jilin Agricultural University, Changchun 130118, China
[2]Jilin Danong Seed Co. , Ltd., Changchun 130000, China

*Corresponding author's email: fengloul@jlau.edu.cn

Abstract

Rice breeding plays a vital role in ensuring global food security; however, effective breeding programs require rapid, accurate, and objective identification of rice diseases. The optimization of rice breeding holds significant importance in achieving increased and high-yield production. Conventional visual inspection methods are labour-intensive, subjective, and prone to human error, particularly when disease symptoms are subtle or visually similar. To address these limitations, this study proposes a non-numerical encoding–based convolutional neural network (NNE-CNN) for rice disease recognition. It expounds upon the algorithm's implementation principles, encompassing aspects such as initialization methods, encoding techniques, and population updating procedures. The performance of the proposed model was evaluated using two public benchmark datasets (Convex and Rectangles) as well as a rice disease image dataset. Subsequently, it utilizes the differential operator to compute an updating operator for the individuals. This process involves the removal or sparsity of certain convolutional layers within the CNN, thereby reducing the model's complexity and computational overhead while preserving its generalization capabilities. By deleting or sparsifying part of the convolutional layers in the CNN, the complexity and computational overhead of the model are reduced, while maintaining the generalization ability of the model. Experimental results demonstrate that NNE-CNN consistently outperforms classical CNN architectures and other intelligent algorithm-optimized models, achieving accuracies of 96.94% and 96.17% on the public datasets and a test accuracy of 97.50% on rice disease images. These findings indicate that the proposed method provides a robust and computationally efficient solution for rice disease identification, with practical implications for automated crop monitoring and data-driven rice breeding optimization.

Key words: Rice disease; Images; Non-numerical encoding; Computer vision

Introduction

The magnitude and complexity of challenges encountered in modern social production continue to increase across diverse domains, including unmanned systems such as robotics, governmental decision-making, large-scale data processing, and biomedical engineering (Li, 2019a; Leonowicz, 2022). Advances in production equipment, mobile terminals, and sensor technologies have resulted in a rapid growth in data generation, thereby highlighting the increasing importance of data analysis in the era of big data. The extraction of invaluable insights from colossal data reservoirs has perpetually remained a focal point of research within the realm of computer science. Particularly noteworthy is the acquisition of information from image data, a domain that has assumed paramount significance with the rapid advancement of surveillance facilities, network video technology, and satellite remote sensing imaging. As a result, image processing technology has emerged as a pivotal application area of machine learning (Chang *et al.,* 2020).

Rice, the predominant staple crop across numerous Asian nations, serves as the dietary foundation for more than 50% of the global population, bestowing it with considerable economic significance. With the rapid growth of the world population, the use of pesticides and bactericidal agents for pest management has become an unavoidable practice to ensure stable and increased rice yields. However, this practice exacts a substantial toll on the environment, concurrently engendering latent perils to the quality and safety of rice produce. Within the context of contemporary rice breeding endeavors, the imperative task at hand involves the precise discernment of breeding quality, the assessment of disease and pest infestations, which assumes paramount importance in the enhancement of crop yields. Both sample data and their analysis stand as pivotal benchmarks in the evaluation of crop quality and the validation of the efficacy of breeding initiatives. Nevertheless, manual inspection methods suffer from inherent limitations, including low efficiency and a high susceptibility to human error and subjective bias. In seed defect identification, visual discrimination among seeds with similar morphological characteristics remains a major challenge, making standardized evaluation procedures difficult to implement (Liu *et al.,* 2022). Although machine vision-based defect detection methods draw upon object features for classification, their inherent limitations are primarily attributed to the fixed image capture positions, the intricacy of image processing, and the impediments to achieving normalization in data processing. In the thousand-grain weight measurement, the image processing method for the sparse state has a high seed recognition rate, but the target detection accuracy for the complex environment with multiple overlapping targets is still low. Consequently, rapid and accurate identification, as well as effective control of rice diseases, remains a task of paramount importance.

Received: 08-08-2025 Revised: 22-10-2025 Accepted: 24-10-2025 Online: 09-01-2026

In the realm of image recognition, traditional image processing and machine learning techniques are often intricate. To address these complexities, hyperspectral and red spectrum technologies are leveraged to capture both spectral and image data from rice disease images (Zhang *et al*., 2021). Conventional machine learning methods, such as discriminant analysis and support vector machines, typically require extensive image preprocessing to extract features such as color, texture, and shape prior to disease classification. This is followed by feature selection procedures aimed at improving recognition accuracy, which further increases methodological complexity.

In the domain of deep learning, Convolutional Neural Networks (CNN) have gained widespread application in image processing and exhibit remarkable efficacy in remote sensing image classification. Owing to the black-box nature of deep neural networks, the selection of an optimal architecture lacks standardized design rules and often relies on empirical expertise and repeated trial-and-error experimentation, which limits further advancement (Zhang *et al.*, 2019). Consequently, the selection of an appropriate neural network architecture presents a formidable NP-hard problem, characterized by exceedingly high complexity. To address this challenge, a suitable approach is required to discern the optimal network structure, thereby enhancing classification accuracy.

Swarm intelligence algorithms are widely recognized for their strong optimization capabilities and have demonstrated considerable potential in addressing complex and high-dimensional problems. Accordingly, this study employs an optimization algorithm to ascertain a more suitable deep neural network structure, optimizing the CNN architecture through non-numerical encoding, with the aim of augmenting image classification accuracy. The main contributions of this study are summarized as follows:

(1) Enhanced rice disease ıdentification: This Study proposes a novel CNN-based approach optimized through non-numerical encoding (NNE-CNN) for autonomous and accurate rice disease identification. The proposed method enhances the ability to recognize and diagnose rice diseases from images, thereby supporting data-driven rice breeding and crop management practices.

(2) Structural optimization of CNN: A differential operator is employed to calculate structural differences between individuals and iteratively refine the CNN architecture. This optimization process enables the removal or sparsification of selected convolutional layers, reducing network complexity and computational cost while preserving generalization performance.

(3) Reduction of computational overhead: The proposed methodology effectively lowers computational requirements by optimizing CNN structure without compromising classification accuracy. This improvement enhances the practical applicability of the model for real-world rice disease recognition and other related computer vision tasks.

Material and Methods

Individual code: This study employs a non-numerical encoding approach to represent structural information within individual particles. . In the design of the proposed method , the focus extends beyond selecting the type of each structural element to encompass detailed super-parameter specifications for each structural component. This encompasses factors like the convolution layer's kernel size, the output feature map dimensions, pool layer dimensions, the quantity of neurons in fully connected layers, and various other pertinent considerations. Each individual can be decoded into some kind of convolutional neural network structure, and the super-parameter information of this layer is retained within each layer structure. Unlike conventional function optimization problems, CNN structure optimization does not require individuals to be expressed in numerical form. Therefore, an appropriate population update mechanism is required when adopting non-numerical encoding.

Individual difference operator of population: Within the non-numerical encoding framework, the process of selecting superior genetic traits from better individuals necessitates the use of a differential operator calculation. This operator computes compute the disparity between two individuals and subsequently determining the update operator based on this difference to enhance the characteristics of individuals. The technique for calculating the difference operator among individuals is illustrated in Fig. 1.

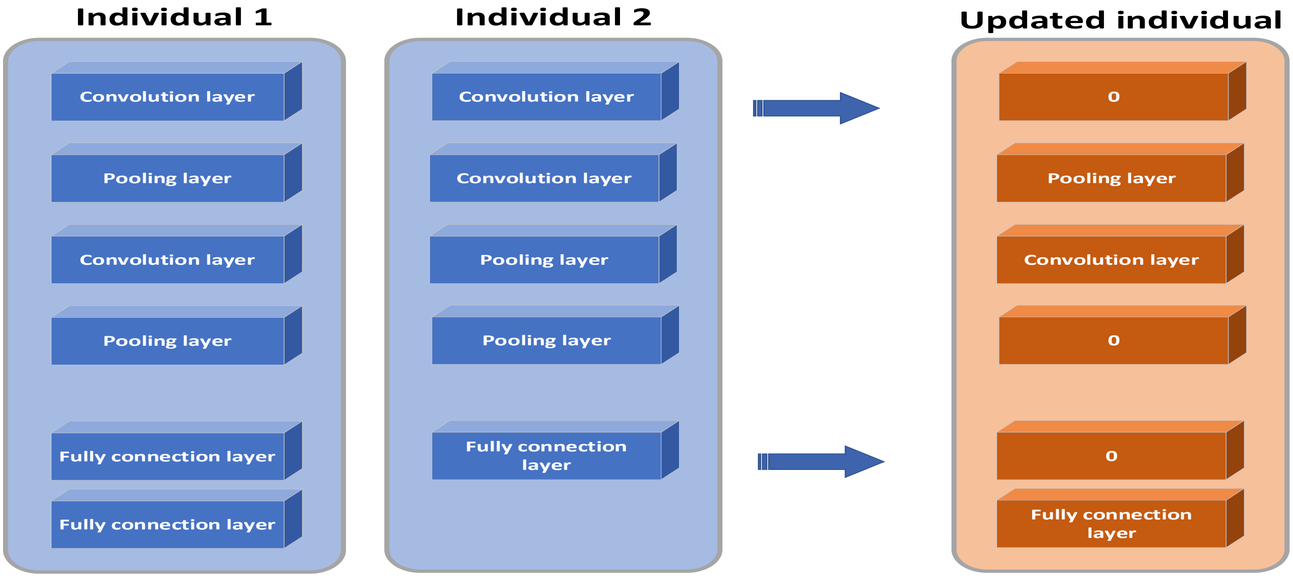

Fig. 1. Individual renewal process.

To ensure the acurate handling of the fully connected layer precise position, it is essential to treat it separately from the other two structural components. When computing the differences between the pool layer and convolution layer, only the structural type differences should be taken into account, while variations in super-parameters should be ignored. For example, if the second layer of two individuals is a convolution layer but with different kernel sizes (e.g., 5 and 3), the super-parameter difference is treated as zero, and the layer position remains unchanged during crossover.

During population hybridization, the stored hyperparameter configuration of the current individual is retained. For instance, during the crossing of sterile lines, if they share the same layer type, such as a convolution layer, but possess varying hyperparameter specifications (e.g., different convolution kernel sizes and diverse feature map dimensions), the original hyperparameters of the sterile lines are retained. During the hybridization process, it is plausible that the number of layers in the network structure for two individuals may not align. In such cases, if the first individual has fewer layers than the second, a '-1' is added to the final difference, signifying the removal of one layer from the structure as a result of crossbreeding. Conversely, if the first individual has more layers, an additional structure is appended to the end of the network.2.3 Population initialization.

The population initialization process generates N individuals with random CNN structures, as detailed in the following algorithm. Each individual is endowed with a random number of layers, ranging from three to the maximum number of layers denoted as *lmax*. This limit applies to all structural components, including convolutional, pooling, and fully connected layers, and prevents uncontrolled network growth. It's crucial to note that the maximum number of layers applies to the entirety of the CNN to prevent an indiscriminate proliferation of network layers.

A standard CNN structure consists of convolution layers, pool layers, and fully connected layers. The fully connected layer performs final feature classification and is therefore typically located at the end of the network. As a result, the fully connected layer typically resides at the end of the convolutional neural network, rather than within its middle layers. Therefore, this algorithm's initialization method takes this into account. Once a fully connected layer is added to the network structure, only additional fully connected layers will be appended until the predetermined number of structural layers is reached. In other words, when an individual adds a fully connected layer, all subsequent layers are fully connected.

The population initialization includes the following steps:

Input: Population size *N*, the preset upper limit of neural network structure layers $lmax$, the upper limit of convolution layer feature maps max, the maximum convolution kernel size k_{max}, the upper limit of the number of neurons in the fully connected layer n_{max}, The number of output results is the category n_{out}.

Output: a population of *N* individuals, , and each representing convolutional neural network (CNN) structure.

Step 1: Set the population size, *N*, and commence the initialization process for each individual.

Step 2: Iterate through the entire population until the entire population has been processed, signifying the completion of the population initialization phase. For each individual, calculate P_iDepth, which represents the current individual, using the Formula (1). This encompasses the total number of CNN structural elements, inclusive of convolution layers, pool layers, and fully connected layers, and then proceed to initialize each individual as described in Step 3.

The formula for finding the maximum number of layers of individual network structure is shown in Formula (1):

$$P_i\ Depth = rand\ (3, l_{max}) \qquad (1)$$

Step 3: Start the traversal from the beginning. If the *PiDepth* is equal to 1, indicating that you have just begun the traversal (i.e., the current layer number is 1), set j to 1. If j is equal to 1, add a convolution layer according to the formulas provided in (2) and (3), and then return to Step 3. If j is not equal to 1, proceed to Step 4.

The convolution layer structure mainly includes the convolution kernel size and the number of filters as shown in Formula (2) and Formula (3) respectively:

$$out\_channel = rand\ (3,\ max\_out\ \_channel) \qquad (2)$$

$$conv\_kernel = rand\ (3,\ conv\ \_\ kernel) \qquad (3)$$

Step 4: If the current layer number, j, corresponds to the last layer (i.e., j equals *PiDepth*), introduce the fully connected layer by applying the formula provided below, and subsequently return to Step 2 to commence the traversal for the next individual. In case j is not the last layer, proceed to Step 5.

Step 5: If the previous layer of the current layer is a fully connected layer, proceed to add another fully connected layer on top of it. Otherwise, advance to Step 6.

The structure of the fully connected layer is mainly the number of neurons, so the formula for adding the whole connective layer is shown in Formula (4):

$$neronsnum = rand\ (1, n_{max}) \qquad (4)$$

When adding a pool layer, the choice between an average pool layer and a maximum pool layer should be determined as follows: Generate a random number, and if it's greater than 0.5, add a maximum pool layer; otherwise, add an average pool layer. The pooling size is fixed at 2x2.

Step 6: Generate a random integer between 1 and 3. When the integer is 1, incorporate a convolution layer. If the integer is 2, introduce a pooling layer. In case the integer is 3, implement a fully connected layer. It is noteworthy that these layers possess distinct probabilities of occurrence, as expounded in this paper, namely 0.6, 0.3, and 0.1. To clarify, the likelihood of incorporating a convolution layer stands at 0.6, the probability of introducing a pooling layer rests at 0.3, and the probability of including a fully connected layer is 0.1. When adding a pooling layer, there is an equal likelihood of integrating either an average pooling layer or a maximum pooling layer, with both options holding a 50% chance, concurrently proceeding to Step 3.

Fitness calculation: The objective of this algorithm is to identify an appropriate convolutional neural network for image structure classification. When assessing the fitness value, each individual undergoes a decoding process to translate it into a meaningful CNN architecture. Subsequently, this architecture is subjected to a brief training phase, during which the model is fine-tuned. Simultaneously, the performance on the test dataset, measured in terms of cross-entropy, is utilized as the fitness value to appraise the quality of the individual.

For a single example, the cross-entropy for that example is calculated as:

$$H(Y, P) = -\Sigma\ [\ Y(i) * \log(P(i))\ ] \qquad (5)$$

In this equation: H(Y, P) is the cross-entropy for the example.

Y(i) is a binary value (0 or 1) indicating whether the actual class is class i.

P(i) is the predicted probability that the example belongs to class i.

The algorithm aims to identify the network architecture with the minimum loss, irrespective of network depth or structural complexity. Nonetheless, this algorithm does possess a drawback, as it necessitates the decoding of each individual into a CNN structure and entails a preliminary training phase, which can be time-consuming. Nevertheless, the strength of this algorithm lies in its potential to discover an optimal structure that may perform well in diverse data collection scenarios, showcasing adaptability across different datasets.

Disease identification based on significance detection: After image I is input, a certain pixel $I_k, I_k \in [0,255]$ in the image is known. After the input image is transformed into grayscale image, histogram D is obtained, and the pixel value A_n is F_n, where $A_n \in [0,255]$. Elements $D(x, y) = |A_x - A_y|$ represents the color of the gap between pixels A_x and pixels A_y, where $A_n \in [0,255], D(x, y) \in [0,255]$.

The pixel I_k significant value in $\mathrm{Sals}\ (I_k)$ can be calculated as formula:

$$\mathrm{Sals}\ (\mathrm{Ik}) = \mathrm{Sals}\ (A_m) = \sum_{n=0}^{255} F_n D(m, n) \qquad (6)$$

To bolster image learning through CNN and augment recognition accuracy, Sals (Ik) denotes the saliency measure associated with the image or element labeled as "Ik." The computed saliency is subtracted from the pixel values of the original image, thereby amplifying contrast and mitigating the impact of intricate backgrounds.

The pixel value of the image after subtraction is Z_k, whose range is [0,255], the calculation process is shown in Formula (7).

$$Z_k = I_k - \mathrm{Sals}\ (I_k) \qquad (7)$$

Following the saliency detection and segmentation of the image, the processed data is fed into the NNE-CNN for training and classification.. To facilitate and streamline the training process of the model, a visual tool, Tensorboard, is employed. Figure 2 illustrates the overall neural network architecture. In the figure, solid lines depict data flow between computational nodes, with arrows indicating the direction of data transmission. The tensor dimensions resulting from the three convolution layers are as follows: 57×57×48, 28×28×128, and 13×13×192. The dimension passed from the final pooling layer to the initial fully connected layer is 6912, and the dimension classified using softmax is 4.

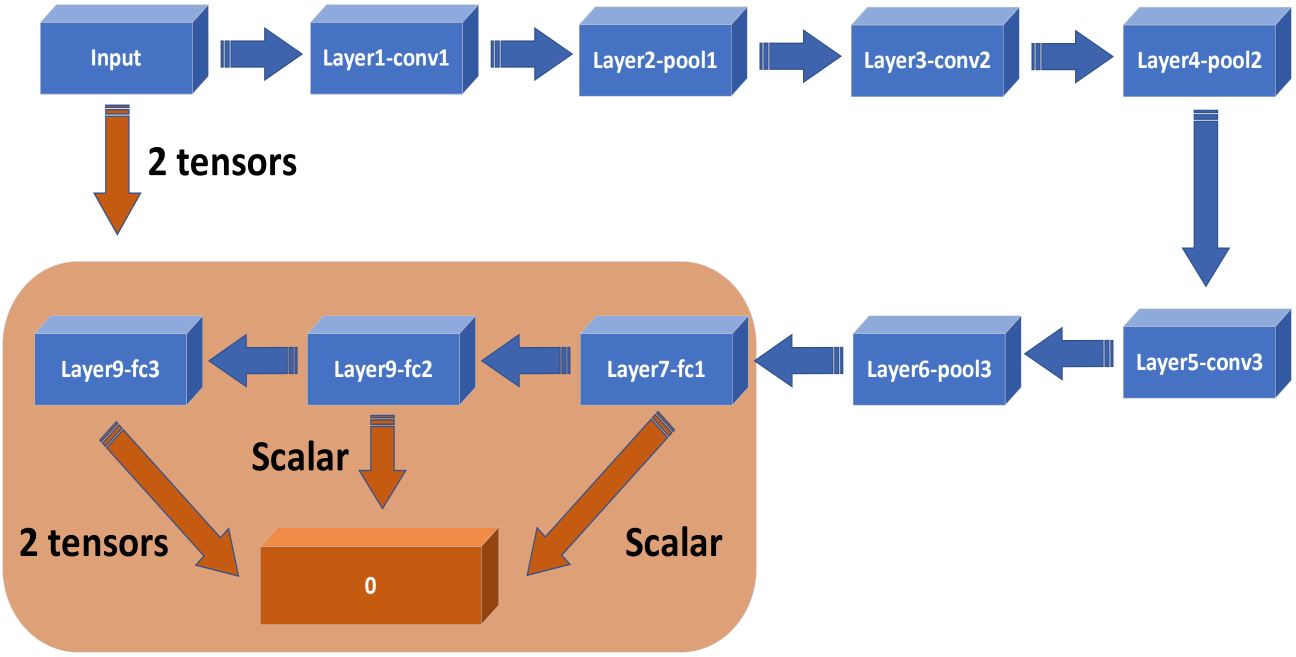

Fig. 2. NNE-CNN structure diagram.

Table 1. Parameter settings.

Parameter	Meaning	Value
N	Number of populations	30
iter	Number of population iterations	10
e_{train}	Number of training sessions per individual	100
lmax	Maximum number of network structure layers	20
lmin	Minimum number of network structure layers	3
max_conv_output_schannel	Upper limit on the number of channels in a convolutional layer	256
k_{max}	Maximum kernel size	7
t	Selection probability of difference operator during hybridization	0.7
e_{train}	Number of training sessions per individual	1
$e_{\text{train\_final}}$	The optimal individual training times	100

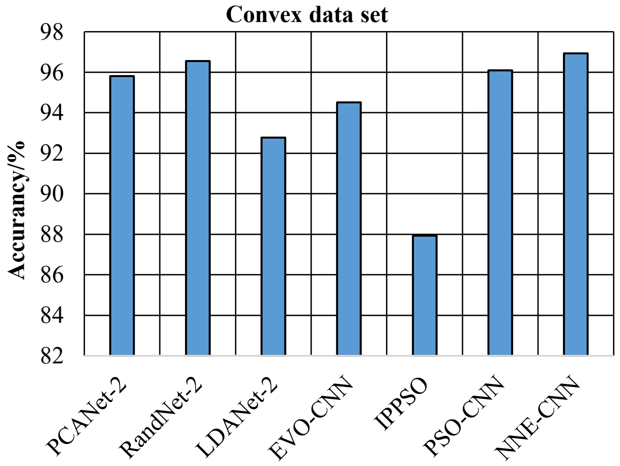

Fig. 3. Classification accuracy comparison of different algorithms on the Convex dataset.

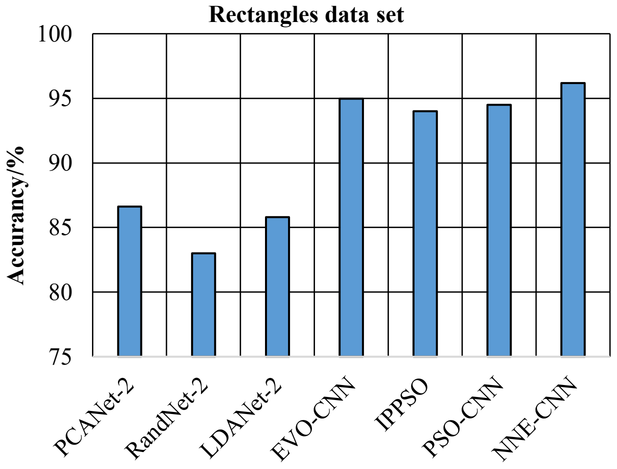

Fig. 4. Classification accuracy comparison of different algorithms on the Rectangles dataset.

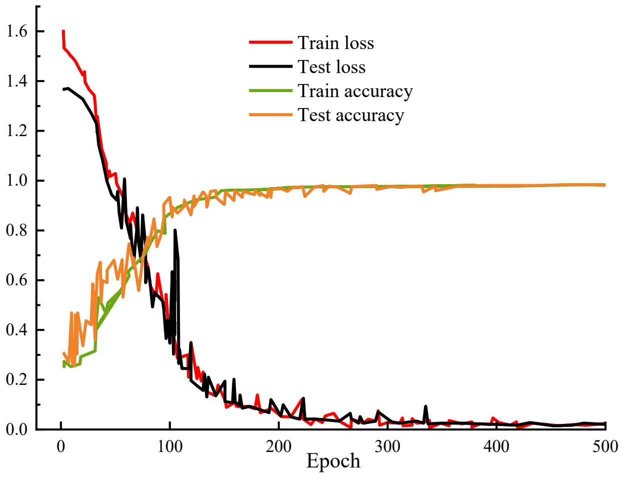

Fig. 5. Training and testing loss and accuracy curves of the NNE-CNN on the rice disease dataset.

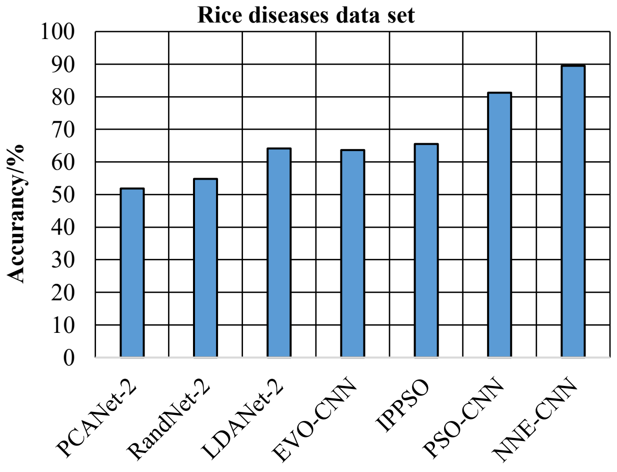

Fig. 6. Classification accuracy comparison of different algorithms on the rice disease dataset.

Results and Analysis

Model verification: Initially, two publicly available datasets, namely Convex and Rectangles-Image, were employed to evaluate the proposed method under consistent experimental conditions. The algorithm's parameter settings are summarized in Table 1.

For computational efficiency, each individual was trained for a single epoch during fitness evaluation, while the optimal architecture was further trained for 100 epochs.

The classification accuracies obtained on the Convex dataset are illustrated in Fig. 3. Owing to the relatively simple structure and sufficient data volume of this dataset, all evaluated models exhibited strong performance. Among the classical approaches, RandNet-2 achieves higher accuracy than PCANet-2 and LDANet-2. However, structure-optimized models demonstrated superior performance overall. In particular, PSO-CNN and the proposed NNE-CNN outperform classical networks, with NNE-CNN achieving the highest accuracy of approximately 96.9%.

As for the public Rectangles dataset, which contains significant background interference, the classification accuracies are presented in Fig. 4. The classical models PCANet-2, RandNet-2, and LDANet-2 all achieved accuracy levels below 90%, indicating limited robustness under complex background conditions. In contrast, structure-optimized algorithms, including EVO-CNN, IPPSO, and PSO-CNN, demonstrated substantially improved performance, with accuracies exceeding 94%. Among all evaluated models, the proposed NNE-CNN achieves the highest accuracy of 96.17%, outperforming both classical approaches and other structure-optimized CNN models.

The results above highlight that PCANet-2, RandNet-2, and LDANet-2 performed well on simpler datasets such as Convex . However, when datasets incorporate interfering elements such as rotation or background images, the performance of these classical models noticeably declines. In such scenarios, the performance of structure optimization algorithms, characterized by their ability to dynamically adapt network architectures, becomes significantly superior to the aforementioned classical models. Consequently, these algorithms demonstrated enhanced performance on complex datasets.

The proposed NNE-CNN exhibits exceptional performance on both Convex and Rectangles datasets. As a result, the paper seeks to apply this algorithm to the recognition of rice disease images.

Application case: To further evaluste the effectiveness of the proposed model in recognizing rice diseases, a dataset was meticulously constructed using images of rice diseases as the target identification objects. To enhance model robustness and feature learning capability, data augmentation techniques were applied to the original images, including random rotations (ranging from 5° counterclockwise to 5° clockwise), random flipping, brightness adjustment, and contrast modification.

The augmented rice disease images were subsequently used to train the NNE-CNN in an end-to-end manner. As shown in Fig. 5, both training and testing loss decreased rapidly during the initial training phase, followed by a gradual stabilization as the number of epochs increases. Although minor fluctuations are observed in the early epochs, the loss and accuracy curves converge smoothly, indicating stable training behavior and good generalization performance. The final training accuracy and test accuracy reach 98.05% and 97.50%, respectively.

The recognition performance of the rice disease image dataset is presented in Fig. 6 in terms of classification accuracy. Among the evaluated models, the proposed NNE-CNN achieves the highest accuracy, outperforming both classical methods and other structure-optimized CNN models. In particular, the classification accuracy of NNE-CNN exceeds that of PSO-CNN by approximately 8.30%, demonstrating its superior performance on rice disease image recognition.

Discussion

Early rice breeding inspection primarily relied on manual selection, a process that is labor-intensive, time-consuming, and constrained by human physical and cognitive limitations. Hence, the development of an automated classification system became imperative to achieve the goals of rapid and accurate recognition. This approach is grounded in non-contact, nondestructive principles, and machine vision emerges as a viable solution. Vanitha employed three deep learning models, namely VGG16, ResNet50, and InceptionV3, to discern rice diseases (Vanitha, 2019). He assembled a comprehensive database of rice diseases, meticulously labeling the images, and leveraging various data augmentation techniques to expand the dataset, his model was designed to identify three distinct leaf diseases, achieving an impressive 99.53% accuracy with the ResNet50 model. Similarly, (Desai *et al.*, 2019) applied deep learning to detect flowering panicles in rice images and estimate heading dates with an average error of less than one day.

Notably, there is a paucity of literature dedicated to deep learning applications in the context of rice, although the advantages of deep learning methodologies have been explored in related agricultural domains. For instance, Ma introduced a prediction method for grain numbers in ears based on deep learning (Ma *et al.*, 2019). Through the utilization of a full convolutional neural network and the precise marking of grain centers, they successfully detected and counted the number of grains on the entire ear without compromising its morphological integrity, achieving a measurement error of 3.47%, in compliance with measurement accuracy standards. Meanwhile, Wang et al., (2023a) enhanced the VGG19 network model by incorporating a global average pooling layer and two dense layers, thereby elevating the recognition accuracy of corn seeds.

Swarm intelligence algorithms, as a subset of intelligent algorithms, offer a valuable approach to approximating solutions for NP problems. The structural optimization of neural networks undeniably constitutes a highly intricate challenge, prompting several scholars to explore the fusion of swarm intelligence algorithms with machine learning or deep learning methods in pursuit of algorithm performance enhancement. Ding *et al.,* (2019) harnessed the power of the particle swarm optimization (PSO) algorithm to identify an optimal set of mapping parameters for extreme learning machines.

Subsequently, these mapped feature samples were fed into the K-Nearest Neighbor (KNN) algorithm, amplifying its capacity to address linear inseparability issues. Pourpanah *et al.*, (2023) leveraged PSO to optimize the structure of Bayesian networks, demonstrating distinct advantages over alternative optimization methods. Yan *et al.*, (2019) devised a hybrid particle swarm optimization algorithm, alongside a quasi-Newton algorithm, to operate on a CPU-GPU platform.

This innovation curtailed data transmission time and reduced training errors simultaneously. Wang *et al.*, (2023b) employed PSO to fine-tune the weights and thresholds of neural networks, successfully applying this approach to estimate and predict the spectrum of soil organic matter content. Furthermore, Huang *et al.*, (2020) harnessed genetic algorithms to optimize feedforward networks for the classification of pre-processed remote sensing images.

In the previously discussed research, the proposed methods were primarily tailored to fully connected neural network structures and weren't directly transferable to the domain of image classification employing deep neural networks. Wang the belief function defined by the composite normal distribution to model the knowledge of experts on the fine-tuning of CNN hyperparameters to enhance the exploration capability of standard PSO(Wang *et al.*, 2019b). In this innovative approach, particle encoding takes inspiration from the IP address coding method in computer networking, with each network layer assigned an IP address. Sun proposed Evolving Deep CNN algorithm which deploys genetic algorithms to explore CNN structures with specific crossover and mutation operators, displaying remarkable performance on numerous public datasets(Sun *et al.*, 2020).

However, it's worth noting that this algorithm demands substantial computational resources. Wang applied a hybrid algorithm combining genetic algorithms and particle swarm optimization to optimize CNN architectures (Wang *et al.*, 2019a). In contrast, Li *et al.*, (2019b) utilized a PSO algorithm based on binary coding to optimize CNN structures. It's essential to recognize that various optimization algorithms come with their unique limitations, and the choice of coding strategy significantly impacts CNN structural optimization. Thus, this paper endeavors to optimize CNN structures via non-numerical encoding, seeking to strike a suitable balance between search accuracy and speed, thereby addressing the challenge of image classification.

The experimental results demonstrate that NNE-CNN exhibits superior performance compared to traditional models and other intelligent algorithm-optimized CNNs across various datasets. On the Convex dataset, NNE-CNN achieved the highest accuracy of 96.94%, while on the more challenging Rectangles dataset with background interference, it attained 96.17% accuracy, significantly outperforming classical models like PCANet-2, RandNet-2, and LDANet-2 which showed accuracy below 90%.

The superior performance of NNE-CNN can be attributed to its non-numerical encoding approach and differential operator mechanism, which enables dynamic adaptation of network structures. This flexibility allows the model to maintain high performance even when dealing with complex datasets containing rotational variations and background noise. The structural optimization through layer removal or sparsification reduces computational overhead while preserving generalization capabilities.

In the rice disease recognition application, NNE-CNN achieved 97.50% test accuracy, demonstrating its practical utility in agricultural applications. The saliency detection-based preprocessing method effectively enhanced contrast and reduced background complexity, contributing to the model's robust performance. The stable training process with minimal fluctuations indicates the reliability of the proposed approach.

However, it's important to note that the algorithm requires substantial computational resources for the initial training phase, which represents a limitation for real-time applications. Future work should focus on optimizing the computational efficiency while maintaining the model's adaptive capabilities.

Conclusion

Swift and accurate identification and detection of rice diseases are of paramount importance, given the critical role rice plays as a staple food crop worldwide, and the significant impact of infections on crop yield. This research employs a non-numerical encoding approach to represent the CNN structure, achieving the hybridization of non-numerical rice seeds through the utilization of difference and update operators.

Through a comparative analysis of the training results on Convex and Rectangles public datasets, the study reveals that NNE-CNN outperforms the PCANet-2, RandNet-2, and LDANet-2 models, especially in challenging data scenarios. The consistent training process of the significance-detection segmented model streamlines the extraction, identification, and classification of rice disease characteristics. This research contributes to the field of agricultural disease identification and proactive disease prevention, offering a theoretical foundation for optimizing crop breeding strategies.

Conflict of Interest: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Authors contribution: Xiao Han, conceptualization, study design, data collection, visualization, writing – original draft; Qingrui Zhang and Ziting Gao, methodology, statistical and data analysis, data curation and organization. Xiaoliang He and Fenglou Ling, validation, writing – revision and editing. All authors have read and approved the final manuscript, underscoring their commitment to the project's quality and integrity.

Acknowledgment / Funding: This study was supported by the following fundingsources: Science and Technology Department of Jilin Province-Natural Science Foundation of Jilin Province (20240101201JC) and Jilin Provincial Department of Science and Technology-Innovation Platform (Base) and Talent Program (20240601061RC).

References

Chang, Z., Z. Du, F. Zhang, F. Huang, J. Chen, W. Li and Z. Guo. 2020. Landslide susceptibility prediction based on remote sensing images and gis: comparisons of supervised and unsupervised machine learning models. *Remote Sens*., 12(3): 502.

Desai, S.V., V.N. Balasubramanian, T. Fukatsu, S. Ninomiya and W. Guo. 2019. Automatic estimation of heading date of paddy rice using deep learning. *Plant Methods*, 15(1): 1-11.

Ding, J., T. Liu and J. Wang. 2019. KNN classification algorithm based on PSO-ELM feature mapping. *Mod. Electron. Technol*., 42(5): 152-156.

Huang, X., J. Shen and G. Li. 2020. Genetic algorithm optimization for remote sensing image classification based on BP neural network. *Mod. Electron. Technol*., 43(12): 47-49.

Leonowicz, Z. and M. Jasinski. 2022. Machine learning and data mining applications in power systems. *Energies*, 15(5): 1676.

Li, K. 2019a. Research on strengthening in-depth study of artificial intelligence in clinical application of medical imaging. *Chin. Med. Imag. Technol*., 35(12): 1769-1770.

Li, Y., J. Xiao, Y. Chen and L. Jiao. 2019b. Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification. *Neurocomputing*, 362: 156-165.

Liu, J., X. Wang and Y. Li. 2022. Deep learning-based detection and classification of rice diseases: A comprehensive review. *Comp. Electron. Agric*., 202: 107373.

Ma, Z., L. Gong and K. Lin. 2019. Estimation of grains in panicle based on pattern recognition of rice panicle geometry. *J. Shanghai Jiaotong Uni*., 396(02): 117-124.

Pourpanah, F., R. Wang, C.P. Lim, X.Z. Wang, D.H. Vu, C. Wang, H. Wang, J. Wang and Ou, Y. 2023. A review of Bayesian neural networks and hybrid Bayesian-based deep learning models: Applications and challenges. *Eng. Appl. Artif. Intell*., 126: 106861.

Sun, Y., B. Xue, M. Zhang and G.G. Yen. 2020. Evolving deep convolutional neural networks for image classification. IEEE Trans. *Evol. Comput*., 24(2): 394-407.

Vanitha, V. 2019. rice disease detection using deep leaming. *Int. J. Recent Technol. Eng*., 7(5S3): 534-542.

Wang, B., Y. Sun, B. Xue and M. Zhang. 2019a. A Hybrid GA-PSO method for evolving architecture and short connections of deep convolutional neural networks. *Swarm Evol. Comp*., 49: 114-123.

Wang, L., H. Wang, J. Liu, C. Li and Y. Zhang. 2023a. MViT: A vision transformer-based framework for fine-grained maize seed variety identification. *Exp. Syst. Appl*., 213: 119000.

Wang, L., J. Wang, X. Li, Y. Zhang and Y. Chen. 2023b. A deep learning approach for soil organic matter content estimation from hyperspectral data using attention-based convolutional recurrent neural networks. *Remote Sens. Environ*., 285: 113382.

Wang, Y., H. Zhang and G. Zhang. 2019b. cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. *Swarm Evol. Comp*., 49: 114-123.

Yan, S., Q. Liu, J. Li and L. Han. 2019. Heterogeneous acceleration of hybrid PSO-QN Algorithm for neural network training. *IEEE Access*, 7: 161499-161509.

Zhang, R., Y. Li, C. Wang, H. Zhang and S. Chen. 2021. Real-time detection of rice blast disease using deep learning with hyperspectral imaging. IEEE Trans. *Geosci. Remote Sens*., 59(12): 10272-10282.

Zhang, W., P. Tang and L. Zhao. 2019. Remote sensing image scene classification using CNN-CapsNet. *Remote Sens*., 11(5): 494.