RICE SEED CLASSIFICATION BASED ON SE-ResNet50
/
基于 SE-ResNet50 的稻种分类

Zhen MA[1], Sa WANG[1], Hongxiong SU[1], Juxia LI*[1], Yanwen LI[1], Zhifang BI[2], Xiaojie LI[1][1]
[1] College of Information Science and Engineering, Shanxi Agricultural University, Taigu 03080/China;
[2] Department of Basic Sciences, Shanxi Agricultural University, Taigu 03080/China
Tel: +8615803446486; E-mail: lijxsn@126.com
Corresponding author: Juxia Li
DOI: https://doi.org/10.35633/inmateh-76-12

Keywords: *rice seed classification, SE-ResNet50 network model, deep learning, Image classification*

ABSTRACT

Traditional rice seed classification methods rely on manual observation of morphological features, which are inefficient and limited in accuracy. To improve the efficiency and accuracy of rice seed classification, this paper proposes a deep learning-based rice seed classification method using the SE-ResNet network architecture. This architecture integrates SENet into ResNet50, enabling the model to capture and learn sensitive differential features among rice seeds. Through comparative experiments, the classification accuracies of SE-ResNet50, ResNet, and AlexNet on the rice seed dataset were 89.58%, 72.97%, and 76.35%, respectively. The results demonstrate that SE-ResNet50 significantly outperforms ResNet and AlexNet in classification accuracy, validating its superiority in rice seed classification tasks.

摘要

传统的稻种分类方法依赖人工经验对形态特征的观察，效率低且准确性有限。为提高稻种分类的效率及精度，本文提出了一种基于深度学习的稻种分类方法，采用 SE-ResNet 网络结构，该结构将 SENet 集成到 ResNet 中，使其能够捕捉并学习稻种之间的敏感性差异特征。通过对比实验，SE-ResNet50、ResNet50 和 AlexNet 三种网络结构在稻种数据集上的分类准确率分别为 89.58%、72.97%和 76.35%。结果表明，SE-ResNet50 的分类准确率明显高于 ResNet50 和 AlexNet，验证了其在稻种分类任务中的优越性。

INTRODUCTION

Rice, as one of the most important global food crops, plays a pivotal role in agricultural production and food security. With the growing global population and increasing demand for food, improving rice yield and quality has become a critical goal in agricultural research. Among these efforts, the classification and identification of rice seeds are key to achieving precision agriculture and variety improvement. Accurate rice seed classification not only aids in seed quality control and variety identification but also provides scientific support for rice cultivation, processing, and trade.

In recent years, with the advancement of computer technology and machine vision, deep learning has gradually been applied to the agricultural domain. Significant progress has been made in rice seed classification. Traditional methods primarily rely on manual observation and morphological feature measurements, such as grain length, width, and shape. However, these methods suffer from subjectivity and inefficiency. In contrast, machine learning and deep learning-based approaches can automatically extract features from rice seed images and achieve efficient classification. *Kamilaris et al., (2018),* systematically compared deep learning with traditional methods and found that deep learning significantly outperforms traditional methods in agricultural tasks, establishing its theoretical superiority. *Carneiro et al., (2024)* systematically analyzed 37 recent studies that employed deep learning (DL) and machine learning (ML) models for grape variety identification, aiming to provide a detailed analysis of the classification process and highlight the strengths and limitations of each method. Based on this, researchers have explored various dimensions of convolutional neural networks (CNNs). *Sladojevick et al., (2016)* and *Lu et al., (2017),* employed basic CNN architectures for plant disease recognition, achieving average accuracies of 96.3% and 95.48%, respectively, demonstrating the versatility of CNNs in complex agricultural scenarios.

[1] *Zhen Ma, Lec.; Sa Wang, M.Sc.Stu.; Hongxiong su,, M.Sc.Stu.; Juxia Li, Prof; Yanwen, Li, Prof.; Zhifeng Bi,Lec.; Xiaojie Li, UG.*

Dymann et al., (2016), and *Reyes et al., (2015),* further expanded the application of CNNs, with the former achieving 86.2% accuracy in classifying 22 plant species and the latter focusing on vein pattern recognition in legumes, both overcoming the limitations of traditional manual feature extraction.

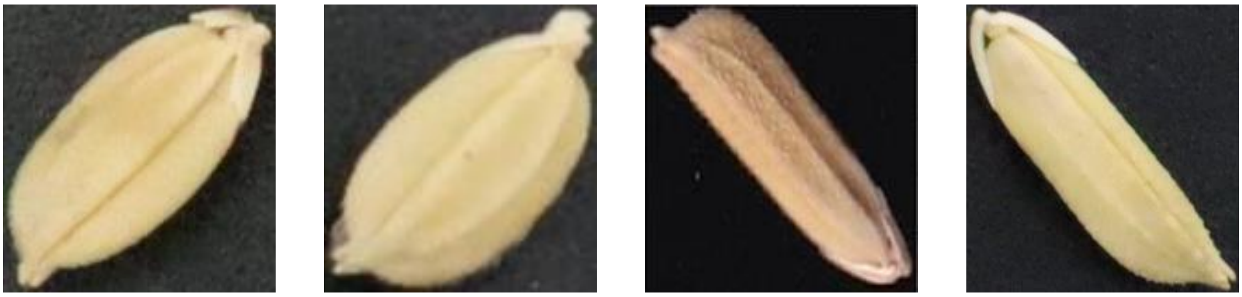
To enhance model performance, researchers have proposed various improvement strategies. *Zhang et al., (2019),* designed the GPDCNN model, which optimizes global pooling and dilated convolution to increase the receptive field while maintaining low computational complexity, significantly improving cucumber disease recognition. *Lu et al., (2017),* visualized feature hierarchies using deconvolution networks, enhancing model interpretability and discriminative power. Meanwhile, transfer learning has become an effective approach to address small-sample problems. *Pereira et al., (2019),* achieved efficient grape variety classification using a fine-tuned AlexNet model, while *Too et al., (2019),* systematically evaluated advanced architectures such as VGG, ResNet50, and DenseNet, finding that DenseNet achieved 99.75% accuracy in classifying 38 plant leaf categories with excellent anti-overfitting properties. *Dong et al., (2024),* addressed the challenge of identifying Camellia oleifera varieties caused by genetic diversity and morphological similarities by constructing a dataset of 30,890 leaf images and developing a deep learning model based on the Convolutional Block Attention Module (CBAM) and RegNetY-4.0GF. *Sun et al., (2023),* improved the EfficientNetv2 network model by introducing a transfer learning mechanism, the adaptive moment estimation optimization algorithm, and the MultiMarginLoss function, addressing the issue of low classification efficiency for southern medicinal plant leaves under complex backgrounds. They achieved a recognition accuracy of 99.12%, representing a 1.17% improvement over the initial model. *Zou et al., (2024*), developed a pepper anthracnose fruit classification and recognition model based on MobileNet V2 and created a mobile-based pepper fruit anthracnose recognition system, addressing the challenge of real-time diagnosis of pepper fruit anthracnose in offline field environments. *Peng et al., (2024*), optimized the YOLOV7 model and proposed the WineYOLO-RAFusion model to achieve more comprehensive and accurate recognition of wine grape varieties. Through data augmentation, transfer learning, and parameter fine-tuning, the MobileNet V2 model achieved a precision rate of 97.31% on the test set, with a recognition time of only 75 ms per image. Although these improved models significantly enhance applicability, they are not well-suited for rice seed classification.

To address these limitations, this paper introduces the SE-ResNet50 model, combining deep learning with feature channel recalibration mechanisms to improve the accuracy and efficiency of rice seed classification. Comparative experiments validate the superiority of SE-ResNet50 in rice seed classification tasks, providing new technical support for agricultural intelligence.

MATERIALS AND METHODS

Dataset Construction

For this experiment, a publicly available dataset of rice seed images was employed. The dataset includes four types of rice seeds: AKITAKOMACHI, KOSHIHIKARI, THAI HOM MALI, and YANGDAO-8, as illustrated in Figure 1. To enhance the generalizability of the model and augment the diversity of the dataset, data augmentation techniques, including random rotation and mirroring, were employed. As a result, the dataset was expanded to a total of 5189 images. The collected seed samples were systematically divided into training and validation sets at an 8:2 ratio, as shown in Table 1.



(a) AKITAKOMACHI (b) KOSHIHIKARI (c) THAI HOM MALI (d) YANGDAO-8

Fig. 1 - Examples of images for each category in the dataset
(a) represents the AKITAKOMACHI class, (b) represents the KOSHIHIKARI class, (c) represents the THAI HOM MALI class, (d) represents the YANGDAO-8 class.

Table 1

Dataset Distribution

Category	Training Set	validation Set
AKITAKOMACHI	976	244
KOSHIHIKARI	1056	263
THAI HOM MALI	1150	287
YANGDAO-8	971	242

Dataset Construction

SE-ResNet50-Based Rice Seed Classification Model

The SE-ResNet50 model proposed in this paper integrates the SE module into ResNet50. The SE module dynamically adjusts the weights of feature channels through Squeeze and Excitation operations, enhancing the network's ability to extract important features.

ResNet50 Overview

ResNet50, or Residual Network, is an innovative deep learning model architecture widely used in computer vision *(He et al., 2019; Veit et al., 2016; Rawat et al., 2017)*. ResNet50's deep network structure (over 100 layers) and exceptional performance have demonstrated its broad applicability in various computer vision tasks. The core innovation of ResNet50 is the introduction of residual learning to address the performance degradation problem caused by increasing network depth. This is achieved through the design of residual blocks, which allow direct information transfer between layers. Each residual block consists of multiple residual units, each comprising two convolutional layers and a skip connection that directly passes the input to the output. This skip connection not only helps maintain network stability but also enhances the network's ability to handle more complex data. The structure of the residual block is shown in Figure 2.

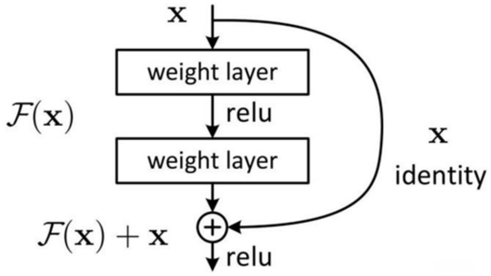

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 2 - Structure of the Residual Block

The diagram summarizes the basic structure of a Residual Network (ResNet50), addressing the vanishing gradient problem in deep neural network training by introducing a skip connection.

The introduction of residual blocks significantly reduces the number of parameters and computational complexity in deep neural networks. Depending on the network depth, residual blocks can take different forms. Typically, ResNet50 employs two main types of residual blocks: the identity mapping module and the downsampling module.

The identity mapping module directly adds the input to the output when the dimensions of the input and output are the same. This process can be mathematically represented as:

$$F_{(x)} = x + H_{(x)} \tag{1}$$

where:

x is the input, $F_{(x)}$ is the output of the residual block, and $H_{(x)}$ is the nonlinear transformation within the residual block. In the identity mapping module, $H_{(x)}$ is typically an identity transformation, that is $H_{(x)} = x$, hence the output $F_{(x)}$ is equal to the input x plus its own value, thus achieving an identity mapping.

The downsampling module is used when the dimensions of the input and output feature maps do not match. This operation reduces the dimensionality of the input feature map to match the output, ensuring the continuity and effectiveness of the network structure. This is typically achieved through convolutional operations. The mathematical representation of the downsampling module is as follows:

$$F_{(x)} = W_{downsample} \cdot x + H_{(x)} \tag{2}$$

where:

x is the input, $F_{(x)}$ is the output of the residual block, $H_{(x)}$ is the nonlinear transformation within the residual block, and $W_{downsample}$ is the weight matrix of the downsampling operation.

These two modules are alternately combined to form the ResNet50 architecture, which is commonly used in constructing deep residual networks. Depending on whether the input and output dimensions are the same, either the identity mapping module or the downsampling module is selected, facilitating the training of deep networks and improving model performance. The structure of ResNet50 is shown in Figure 3.

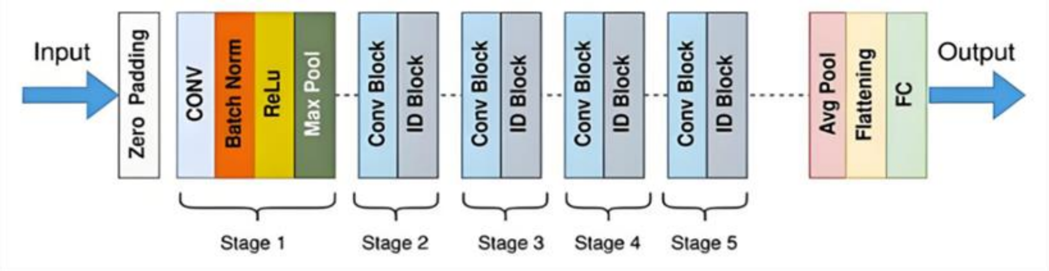


Fig. 3 - Structure of ResNet50

Figure 3 illustrates the overall structure of the ResNet50 model, detailing how data is processed at each stage of the network. The initial stage includes zero padding, a convolutional layer (CONV), batch normalization (Batch Norm), ReLU activation, and a max pooling layer (Max Pool). These layers are used to extract preliminary features from the image and reduce the size of the feature map. In stages 2 to 5, multiple residual blocks, including "Conv Block" and "ID Block," are used to capture deeper features while addressing the vanishing gradient problem through skip connections, thereby improving the network's learning efficiency. Each stage typically begins with a "Conv Block" to adapt to changes in feature map size and reduce computational complexity through bottleneck design. This is followed by several "ID Blocks" for further feature extraction without altering the feature map size. The network ends with a global average pooling layer (Avg Pool) to reduce the feature map to a fixed-size vector. This is followed by a flattening operation to convert the pooled output into a one-dimensional array. Finally, a fully connected layer (FC) maps the extracted features to the probabilities of each class.

Introduction of the SENet-Block Module

The SENet-block module integrates the Squeeze-and-Excitation (SE) module into traditional convolutional neural network blocks *(Hu et al., 2018; Wang et al., 2017; Vaswani et al., 2017)*. This integration enhances the feature representation capability of the network by dynamically adjusting the importance of feature channels, allowing the network to focus more on useful information. Although the SENet-block module may vary across different network architectures, the core idea remains the same: adding the SE module after traditional CNN blocks. The differences between rice seeds are often subtle, and with the development of hybrid rice varieties, these differences have become even more blurred, making the classification of rice seed varieties increasingly challenging. To address this issue, the introduction of the SENet-block module has proven to be an effective solution. The structure of the SENet-block module is shown in Figure 4.

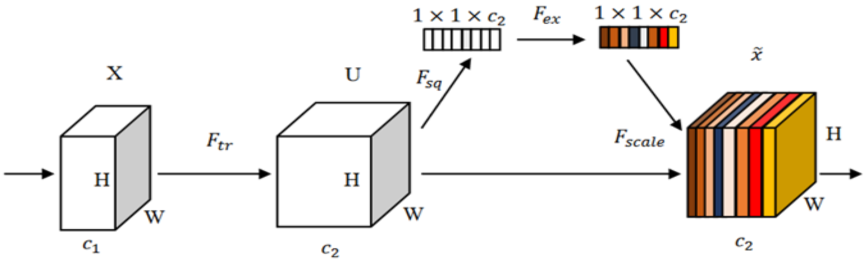


Fig. 4 - Structure of the SENet-Block Module

Figure 4 illustrates the workflow of the SENet (Squeeze-and-Excitation Network) module. First, the input feature map undergoes a Squeeze operation, which compresses the spatial information of each channel through global average pooling. Next, the Excitation operation generates weights for each channel through two fully connected layers. Finally, the Scale operation multiplies these weights with the original feature map to produce the recalibrated output feature map. This process dynamically adjusts the channel weights, enhancing the network's ability to focus on important features and thereby improving model performance.

Squeeze Operation

Through global average pooling, the feature map U is compressed, merging the $H \times W$ spatial information of each channel into a single value, generating a $1\times1\times C_2$ channel descriptor. The purpose of this step is to extract global information from each channel.

The formula for global average pooling is:

$$\mathrm{s_c} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{3}$$

where:

S_c is the output feature of channel C, $H \times W$ is the spatial dimension of the feature map U, $\sum_{i=1}^{H} \sum_{j=1}^{W}$ indicates the summation of all spatial positions (i,j) in the feature map U, $u_c(i,j)$ is the value of the feature map U at channel C and position (i,j).

Excitation Operation

The channel descriptor is first processed by two fully connected layers. The first fully connected layer typically reduces the dimensionality and applies a ReLU activation function. The second fully connected layer maps the dimensionality back to the original number of channels C_2 and uses a sigmoid function to output the weight for each channel. This can be expressed as:

$$z = \sigma(g(s, W)) = \sigma(W_2 \times \delta(W_1 s)) \tag{4}$$

where:

s is the compressed channel descriptor obtained from the Squeeze operation, W_1 and W_2 are the weight matrices of the two fully connected layers, δ represents the ReLU activation function, σ is the sigmoid activation function, and z is the weight for each channel.

Scale Operation

The weights output from the Excitation step x (one weight per channel) are multiplied with the corresponding channels of the original feature map U. This step enhances the channels that the model deems important and suppresses the less important ones. The formula for this step is as follows:

$$\tilde{x}_\mathrm{c} = z_\mathrm{c} \cdot u_\mathrm{c} \tag{5}$$

where:

$\tilde{x}_c$is the recalibrated feature map for channel C, z_c is the weight for channel C, and U_c is the original feature map for channel C.

The above content details the complete structure and operational process of the SENet-block module. By embedding the SENet-block structure into ResNet50, the advantages of deep residual networks addressing the vanishing gradient problem through skip connections—are combined with the feature channel recalibration strategy of SENet-block, significantly enhancing the overall performance of the network. This integration not only preserves the residual learning advantages of ResNet50 but also optimizes the network's ability to process feature information through the feature channel recalibration mechanism, thereby effectively improving network performance.

SE-ResNet50 Network Structure

SE-ResNet50 is a convolutional neural network architecture that integrates the Squeeze-and-Excitation (SE) module into ResNet50. This structure enhances the expressive power and performance of ResNet50 by adding a channel attention mechanism. By introducing the SE module, the network's ability to represent features is improved without significantly increasing computational cost, particularly in image classification and other visual tasks. The SE-ResNet50 model retains the advantages of the original ResNet50, such as using skip connections to train deep networks, while also enhancing the emphasis on channel-level features through the integration of the SE module. This makes the network more effective in handling complex images or other high-dimensional data.

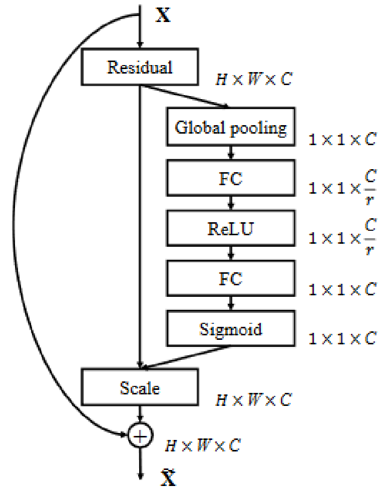


Fig. 5 - Structure of the SE-ResNet50 Residual Block

Figure 5 illustrates the working principle of the SE module combined with skip connections. First, the input feature map X (with dimensions $H \times W \times C$) is compressed through global average pooling into a channel descriptor of size $1 \times 1 \times c$, extracting global information from each channel. Next, the descriptor is processed by two fully connected layers (FC): the first layer reduces the dimensionality to $C \div r$ and applies a ReLU activation function to introduce nonlinearity; the second layer restores the dimensionality to C and generates weights for each channel using a sigmoid activation function. Finally, these weights are multiplied with the original input feature map through a scaling operation, producing the recalibrated feature map x which retains the original dimensions $H \times W \times C$. This process dynamically adjusts the channel weights, enhancing the network's ability to focus on important features and thereby improving model performance without altering the feature map dimensions.

Rice Seed Classification System

To enhance the usability and operational efficiency of the algorithm, this study has developed a rice seed classification application, realized through a dedicated graphical user interface (GUI). Utilizing an object-oriented programming approach, the Qt framework has been employed for the design of this application. The Qt framework facilitates comprehensive interface design by instantiating objects, configuring their properties, and establishing event bindings. This not only achieves separation of the front-end and back-end but also enables real-time target detection. The framework provides an extensive suite of GUI development tools, encompassing over 620 classes and nearly 6,000 functions and methods.

During the operation, users can import the images of seeds that need to be classified into the system by clicking the "Upload Image" button. Subsequently, by clicking the "Start Detection" button, the system will automatically execute the classification process, as illustrated in the figure 6.

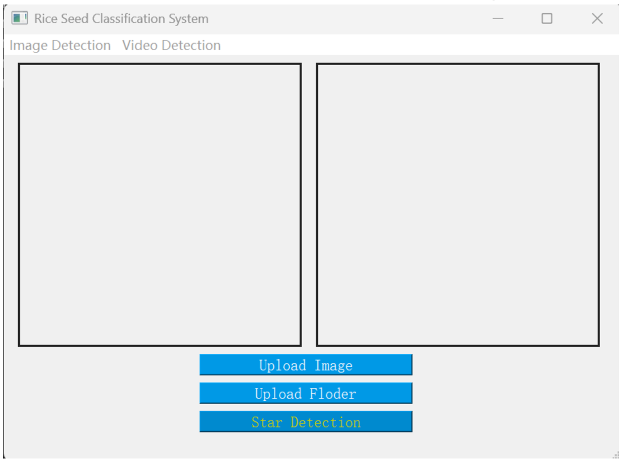


Fig. 7 - Rice seed classification system

Network Model Training

Experimental Platform and Training Parameters

The experimental environment runs on Windows 10 (64-bit) with 16 GB of RAM, an NVIDIA GTX1650Ti GPU, and an Intel(R) Core(TM) i5-10200H CPU @ 2.40GHz processor. The programming platform is Anaconda 4.12.0, with CUDA 10.1, and the development environment is PyTorch, using Python 3.8.

All models in this experiment were trained on the same training set with identical parameters for rice seed classification. The number of epochs was set to 100, the learning rate was 0.0125, and the batch size was 16, ensuring the fairness and comparability of the experimental results.

Model Evaluation Metrics

This method provides four key metrics—True Positives, False Positives, True Negatives, and False Negatives—to reveal the model's performance across different categories, offering a comprehensive understanding of its effectiveness.

Based on these four metrics, more complex evaluation metrics can be derived, such as precision, recall, and F_1 score. $\mathrm{Pr}ecision$ represents the ratio of correctly predicted positive instances to all predicted positive instances. $\mathrm{Re}call$ reflects the ratio of correctly predicted positive instances to all actual positive instances. The F_1 score is the harmonic mean of precision and recall, providing a balanced measure between the two. These evaluation metrics collectively form a comprehensive framework for assessing model performance. The formulas for these metrics are as follows:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$\mathrm{Recall} = \frac{TP}{TP+FN} \tag{7}$$

$$F_1 = \frac{2\times Precision\times \mathrm{Recall}}{Precision+\mathrm{Recall}} \tag{8}$$

For multi-class classification, $\mathrm{Precision}$ is calculated as the ratio of true positives for each class to the sum of true positives and false positives for that class. Recall is calculated as the ratio of true positives for each class to the sum of true positives and false negatives for that class. The overall evaluation metrics are as follows:

$$\mathrm{Precision}_{\mathrm{micro}} = \frac{\sum_{i=1}^{n}\mathrm{TP}}{\sum_{i=1}^{n}\mathrm{TP}+\sum_{i=1}^{n}\mathrm{FP}} \tag{9}$$

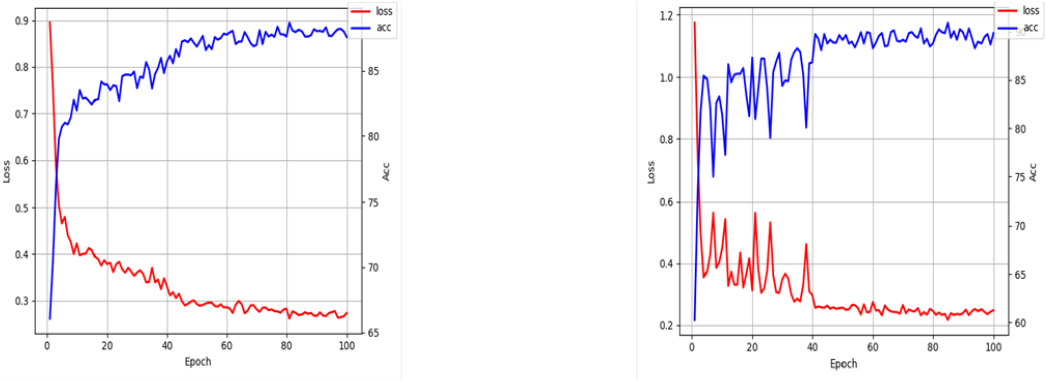
$$\mathrm{Recall}_{\mathrm{micro}} = \frac{\sum_{i=1}^{n}\mathrm{TP}}{\sum_{i=1}^{n}\mathrm{TP}+\sum_{i=1}^{n}\mathrm{FN}} \tag{10}$$

$$\mathrm{F1}_{\mathrm{micro}} = \frac{2\times \mathrm{Precision}_{\mathrm{micro}}\times \mathrm{Recall}_{\mathrm{micro}}}{\mathrm{Precision}_{\mathrm{micro}}+\mathrm{Recall}_{\mathrm{micro}}} \tag{11}$$

RESULTS AND ANALYSIS

SE-ResNet50 Model

The accuracy and loss curves of the SE-ResNet50 model during training and testing are shown in Fig.7.



(a) Training set results

(b) Testing set results

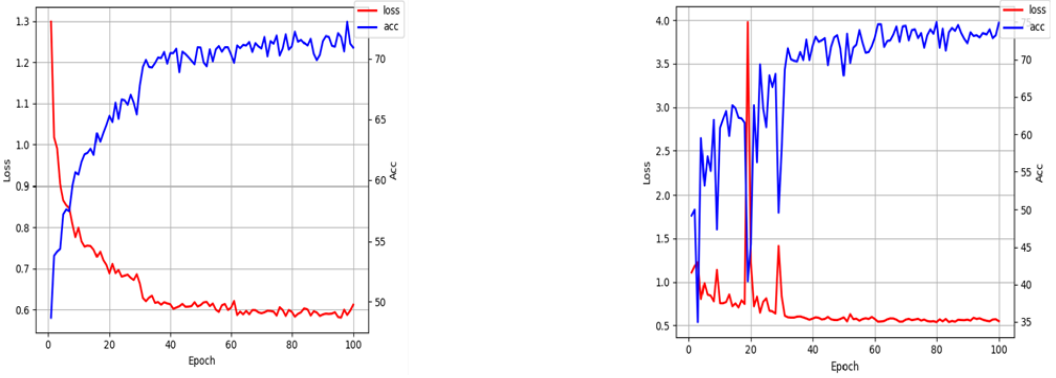
Fig. 7 - Accuracy and Loss Curves of the SE-ResNet50 Model

Figure 6 shows the loss and accuracy curves of the SE-ResNet50 model during training and testing. On the training set, the loss (blue curve) initially decreases rapidly and stabilizes around 0.3 after 100 epochs, while the accuracy (red curve) increases quickly and stabilizes around 95%. On the testing set, the loss (blue curve) also decreases rapidly initially but exhibits more fluctuations, stabilizing around 0.4. The accuracy (red curve) on the testing set increases quickly but fluctuates more, stabilizing around 85%.

From Figure 6, it can be observed that the SE-ResNet50 model exhibits excellent convergence performance during training, with the loss decreasing rapidly and stabilizing. The model shows no signs of overfitting or underfitting, indicating good generalization ability. When applied to the classification task of four types of rice seed images, the model achieves an accuracy of 89.58% on the testing set.

ResNet50 Model

The accuracy and loss curves of the ResNet50 model during training and testing are shown in Fig. 8.



(a) Training set results (b) Testing set results

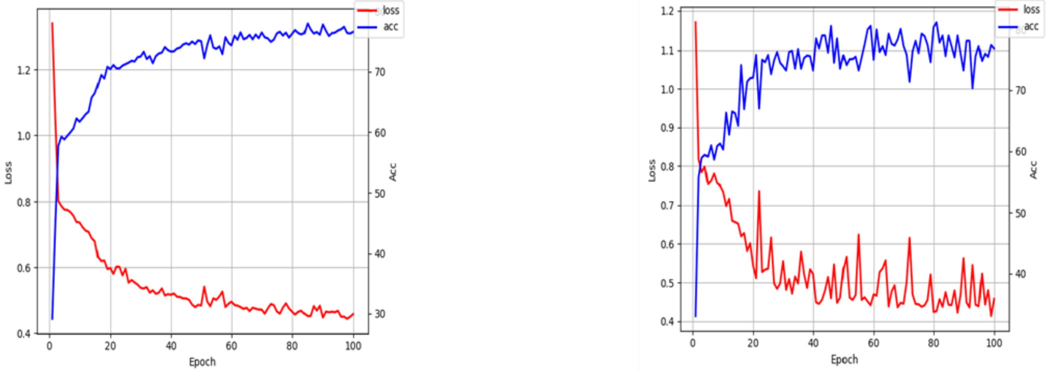
Fig. 8 - Accuracy and Loss Curves of the ResNet50 Model

Figure 7 shows the loss and accuracy curves of the ResNet50 model during training and testing. On the training set, the loss (blue curve) decreases rapidly initially and stabilizes around 0.4 after 100 epochs, while the accuracy (red curve) increases quickly and stabilizes around 70%. On the testing set, the loss (blue curve) decreases rapidly initially but exhibits more fluctuations, stabilizing around 0.5. The accuracy (red curve) on the testing set increases quickly but fluctuates more, stabilizing around 60%.

From Figure 7 it can be observed that the ResNet50 model converges more slowly and fails to achieve the expected performance level. The model's performance on both the training and testing sets is suboptimal. When applied to the classification task of four types of rice seed images, the ResNet50 model achieves an accuracy of only 72.97%.

AlexNet Model

The accuracy and loss curves of the AlexNet model during training and testing are shown in Figure 9.



(a) Training set results (b) Testing set results

Fig. 9 - Accuracy and Loss Curves of the AlexNet Model

Figure 8 shows the loss and accuracy curves of the AlexNet model during training and testing. On the training set, the loss (blue curve) decreases rapidly initially and stabilizes around 0.4 after 100 epochs, while the accuracy (red curve) increases quickly and stabilizes around 70%.

On the testing set, the loss (blue curve) decreases rapidly initially but exhibits more fluctuations, stabilizing around 0.5. The accuracy (red curve) on the testing set increases quickly but fluctuates more, stabilizing around 60%.

From Figure 8, it can be observed that the AlexNet model also converges more slowly and fails to achieve its best performance. The model's performance on both the training and testing sets is suboptimal. When applied to the classification task of four types of rice seed images, the AlexNet model achieves an accuracy of 76.35%.

Comparison of Different Networks

To validate the impact of different networks on model accuracy, this study conducted experiments using SE-ResNet50, ResNet50, and AlexNet. The comparison results are shown in Table 2.

Table 2

Comparison of Experimental Results

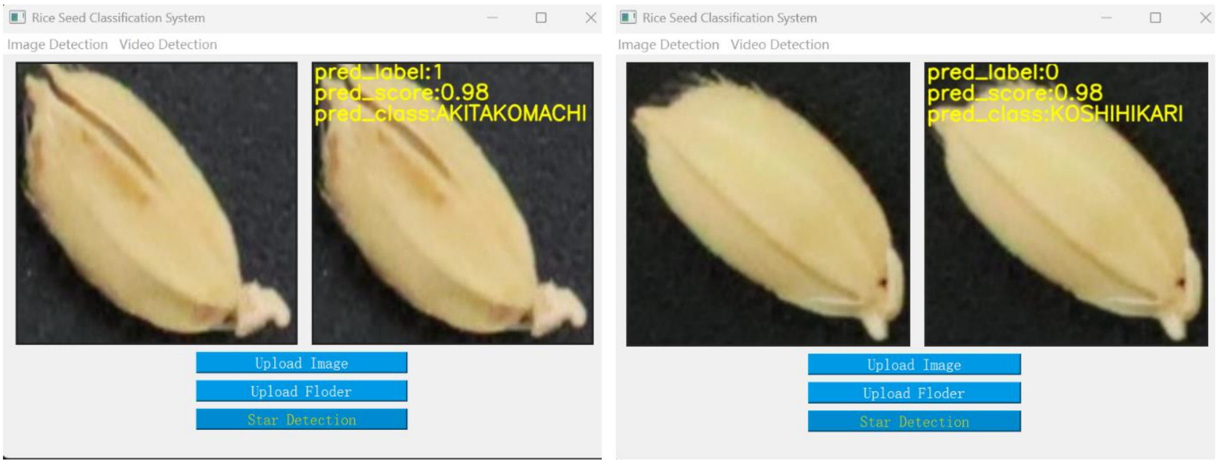
Model	Precision	Recall	F_1 Score
SE-ResNet50	0.8950	0.8922	0.8928
ResNet50	0.7212	0.7202	0.7194
AlexNet	0.7484	0.7582	0.7511

Table 2 shows the performance comparison of SE-ResNet50, ResNet50, and AlexNet in the rice seed image classification task. SE-ResNet50 achieves the best performance in precision (0.8950), recall (0.8922), and F1 score (0.8928), significantly outperforming ResNet50 (precision: 0.7212, recall: 0.7202, F1 score: 0.7194) and AlexNet (precision: 0.7484, recall: 0.7582, F1 score: 0.7511). These results demonstrate that SE-ResNet50 achieves significantly higher accuracy in rice seed image classification compared to the unmodified ResNet50 and AlexNet networks, highlighting the superiority of the selected network model in this task. Specifically, the deeper network layers and feature recalibration strategy (i.e., SENet) of SE-ResNet50 enable it to achieve better classification results in fine-grained image classification tasks such as rice seed classification.

These experimental results comprehensively and convincingly demonstrate that SE-ResNet50 offers higher accuracy and stability in rice seed image classification, validating its significant advantages in classification performance.

Results Presentation of the Rice Seed Classification System

Rice Seed Classification System visually presents the application effectiveness of the constructed intelligent classification model through its user interface (UI), emphasizing the classification process and output results of test set samples. The core interface design focuses on structured output presentation, including (pred_label, pred_class, pred_score), as illustrated in Fig.10. Experimental results demonstrate that the system effectively validates model performance. By visually correlating confidence scores (pred_score) with variety labels (pred_class), it provides an intuitive assessment basis for evaluating the practical applicability and classification accuracy of the model in real-world scenarios. This capability further facilitates subsequent model optimization and system deployment decisions.

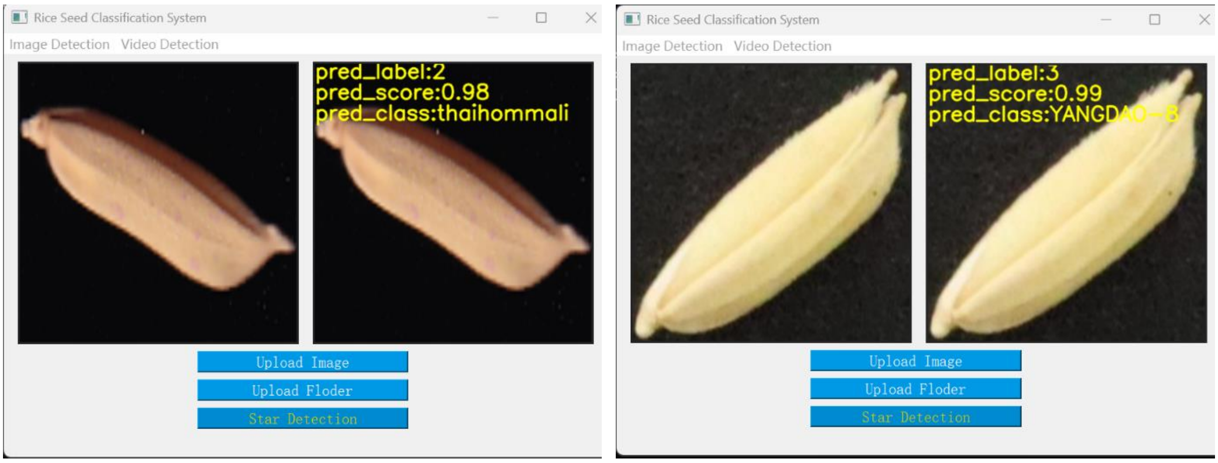


pred_label:1
pred_score:0.98
pred_class:AKITAKOMACHI

pred_label:0
pred_score:0.98
pred_class:KOSHIHIKARI

(a) Classification results for AKITAKOMACHI variety

(b) Classification results for KOSHIHIKARI variety


Rice Seed Classification System
Image Detection Video Detection
Upload Image
Upload Floder
Star Detection
Rice Seed Classification System
Image Detection Video Detection
Upload Image
Upload Floder
Star Detection

```
pred_label:2
pred_score:0.98
pred_class:thaihommali
```

```
pred_label:3
pred_score:0.99
pred_class:YANGDAO-8
```

(c) Classification results for Thai Hom Mali variety (d) Classification results for YANGDAO-8 variety

Fig. 10 - Test Result Chart

CONCLUSIONS

This study innovatively proposes an SE-ResNet network architecture and develops a rice seed classification system, effectively addressing the technical limitations of traditional methods and existing deep learning models in fine-grained agricultural image recognition. The main contributions include:

(1) The novel integration of Squeeze-and-Excitation (SE) blocks into ResNet50 architecture significantly enhances the model's discriminative feature extraction capability for morphologically similar rice seeds through channel attention mechanisms. The proposed system achieves outstanding performance with precision of 0.8950, recall of 0.8922, and F1-score of 0.8928 on a dataset comprising 5,189 images across four varieties.

(2) A Qt framework-based graphical user interface (GUI) application system was developed, enabling real-time rice seed classification and structured output presentation (including category code pred_label, variety name pred_class, and confidence score pred_score), providing an intelligent tool for seed quality control and breeding decision-making.

These research outcomes offer an effective technical solution for agricultural intelligence development. Future studies will focus on expanding the system's applications in smart agriculture through the incorporation of multimodal sensor data and model lightweight design.

ACKNOWLEDGEMENTS

This work was supported by fundamental research program of Shanxi province) (Project No. 202303021212120).

REFERENCES

[1] Carneiro, G. A., Cunha, A., Aubry, T. J., & Sousa, J. (2024). Advancing Grapevine Variety Identification: A Systematic Review of Deep Learning and Machine Learning Approaches. *AgriEngineering*, 6(4), 4851-4888. https://doi.org/10.3390/agriengineering6040277

[2] Dong Z, Yang F, Du J, et al., (2024), Identification of varieties in Camellia oleifera leaf based on deep learning technology [J]. *Industrial Crops and Products*, 216: 118635.

[3] Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*, 151, 72-80.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp.770-778. https://doi.org/10.1109/CVPR.2016.90

[5] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 7132-7141. https://doi.org/10.1109/CVPR.2018.00745.

[6] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70-90. https://doi.org/10.1016/j.compag.2018.02.016

[7] Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice varieties using deep learning. *Journal of Integrative Agriculture*, 16(11), 2516-2525.

[8] Peng J, Ouyang C, Peng H, Hu W, Wang Y, Jiang P. MultiFuseYOLO: Redefining Wine Grape Variety Recognition through Multisource Information Fusion. *Sensors* (Basel). 2024 May 6;24(9):2953.

[9] Pereira, C. S., Morais, R., & Reis, M. J. C. S. (2019). Deep learning techniques for grapevine variety identification. *Computers and Electronics in Agriculture*, *164*, 104918.

[10] Rawat, W., & Wang, Z. (2017). A comprehensive survey of deep learning for image classification. *Neural Computing and Applications*, *28*(10), 3037-3062. https://doi.org/10.1007/s00521-016-2487-1

[11] Reyes, A. K., Caicedo, J. C., & Camargo, J. E. (2015). Fine-tuning deep convolutional networks for plant recognition. *InCLEF* (Working Notes).

[12] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, *2016*, 3289801. https://doi.org/10.1155/2016/3289801

[13] Sun Daozong, Liu Jinyuan, Ding Zheng, et al., (2023), A Multi-Variety Classification Method for Southern Medicinal Plant Leaves Based on the Improved EfficientNetv2 Model [J] (基于改进 EfficientNetv2 模型的多品种南药叶片分类方法). *Journal of Huazhong Agricultural University*, 42(1): 258-267.

[14] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*, 272-279. https://doi.org/10.1016/j.compag.2018.03.032

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS).* pp.5998-6008.

[16] Veit, A., Wilber, M. J., & Belongie, S. (2016). Residual networks: A survey. *arXiv preprint arXiv*:1603.05027.

[17] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., ... & Tang, X. (2017). Attention mechanisms in computer vision: A survey. *arXiv preprint arXiv:1711.07971*.

[18] Zhang, S., Zhang, S., Zhang, C., Wang, X., & Shi, Y. (2019). Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Computers and Electronics in Agriculture*, *162*, 422-430. https://doi.org/10.1016/j.compag.2019.03.012

[19] Zou Wei, Yue Yanbin, Feng Enying, et al. (2024), Identification of Anthracnose Disease in Chili Fruits Based on MobileNet V2 and Its Application [J] (基于 MobileNet V2 的辣椒果实炭疽病识别及其应用). *Guizhou Agricultural Sciences*, 52(09): 125-132.