Journal of Electronic Research and Application, 2026, Volume 10, Issue 2
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

GRIN Lens End Face Classification Detection Based on Deep Learning

Jianqiang Zhang, Yong'an Fu, Foxiang Zhang

Guohua (Hami) New Energy Co., Ltd., Hami 839000, Xinjiang, China

Abstract: With the rapid development of optical communication technology, the demand for gradient-index (GRIN) lenses has increased significantly, making quality inspection of lens end faces an increasingly critical issue. In particular, accurate detection of defects on both ends of GRIN lenses remains a challenging task. To address this problem, this study employs a transfer learning-based parameter fine-tuning approach to evaluate the classification performance of four deep learning models on a defect dataset. Among the evaluated models, ResNet50 and DenseNet-169 demonstrated superior performance and were selected for further optimization. Attention mechanisms, including squeeze-and-excitation (SE) and convolutional block attention module (CBAM), were incorporated into these models to enhance feature representation. Experimental results show that, after integrating the SE module, the classification accuracy of ResNet50 and DenseNet-169 increased by 0.0243 and 0.0272, respectively. With the addition of the CBAM module, the accuracy improvements reached 0.0437 for ResNet50 and 0.0506 for DenseNet-169. These results indicate that the proposed improvements significantly enhance the defect detection capability of the models. All evaluation metrics show consistent improvement over the baseline models, demonstrating that the integration of attention mechanisms effectively increases the classification accuracy and overall performance of the original network architectures.

Keywords: Transfer learning; Classification model; Attention mechanism

Online publication: April 3, 2026

1. Introduction

With the widespread application of fiber optics and lasers, the optical communication industry is developing rapidly, and the use of GRIN lens is gradually increasing [1]. In actual production, quality defects such as edge chipping and scratches may occur on the end face of GRIN lens. However, due to their small size and transparent material, strict requirements are placed on the construction of visual inspection environments. Currently, quality inspection mainly relies on the experience of quality inspectors and is completed under optical equipment, which is difficult, inefficient, and prone to errors, leading to problems such as false inspection and missed inspection [2,3].

The quality of the surface of optical components has an important impact on the performance of optical

systems. Common surface defects of optical components include surface damage such as scratches and pits, as well as residual pollutants such as dust and fibers. They all reduce the service life and load capacity of optical components. In terms of non-destructive testing and recognition of surface defects, the main methods include visual inspection, weighing method, and machine vision based detection [4–8].

This article focuses on the classification problem of GRIN lens end defects, and compares the classification performance on VGG Net, Res Net, Mobile Net, and Dense Net datasets using transfer learning methods. The selected ResNet50 and DenseNet169 models not only have high classification accuracy on the dataset, but also have shorter model runtime in terms of time. Furthermore, a comparative experiment was conducted on ResNet50 and DenseNet169 before and after adding the SE attention mechanism module to the dataset, respectively. In addition, the fully connected layer within the attention module was replaced with a convolutional layer using a 1×1 kernel. Meanwhile, the final classification stage was modified by adopting a global average pooling layer as the classifier instead of a traditional fully connected layer.

2. Dataset introduction

2.1. Introduction to GRIN lens

This article focuses on typical GRIN lenses. As shown in **Figure 1**, it is a transparent cylinder with a diameter of 1.8 mm and a height of 4.75 mm. The two ends of the lens have a circular cross-section on one side and a small angled oblique section on the other side（θ = 8°), a = 0.5 mm, Pitch is 1/4P.

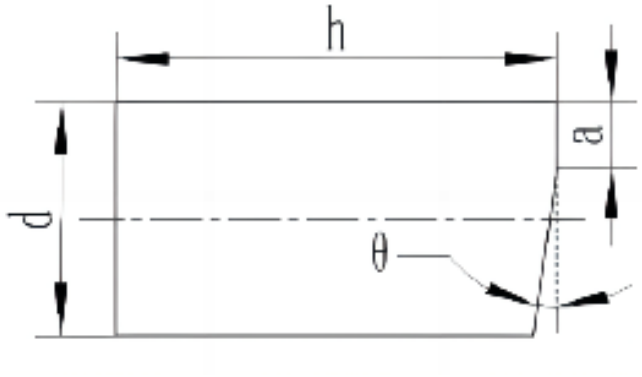

Figure 1. Structure diagram of GRIN lens.

2.2. End face image dataset

During the image acquisition process of the GRIN lens end face, two types of images will appear. The GRIN lens shown in **Figure 1** has a small angle cut at the right end. In **Figure 2**, a clear tangent is presented, dividing the image into two parts. These types of images are named front and encoded as 0. In **Figure 2 (a)**, there is a large halo that differs significantly from the front image in terms of features. This type of image is named the opposite and encoded as 1. As shown in **Table 1**, 488 front and 540 opposite images were collected at any time. As shown in **Figure 3**, **Figure 4**, **Figure 5**, **Figure 6** and **Figure 7**, after preliminary collection of a certain number of images, five data enhancement operations were randomly performed on the images, including brightness enhancement, rotation, image quality improvement, flipping, and noise removal, to further expand the data volume.

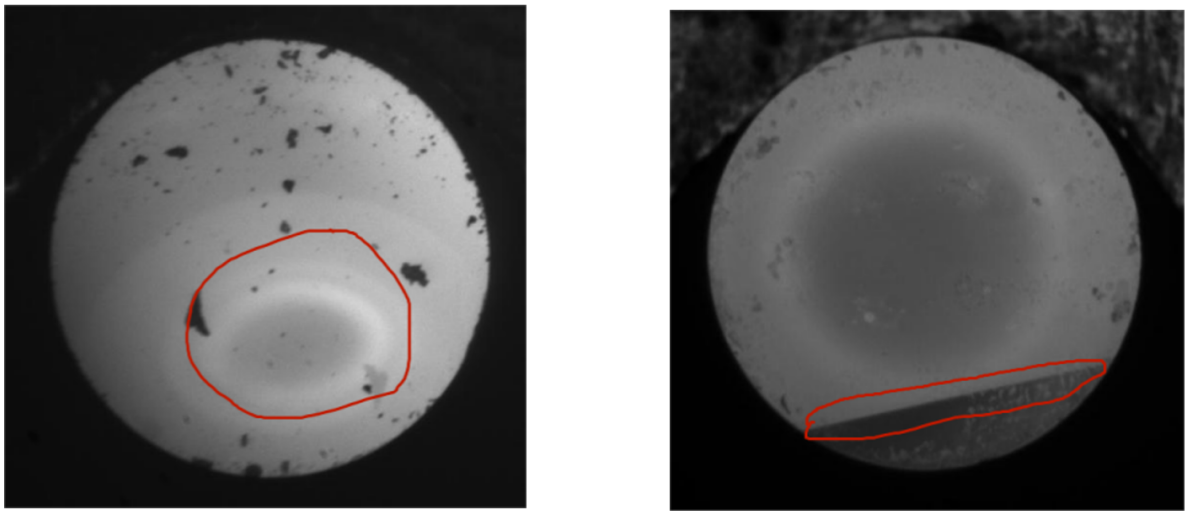

Figure 2. End face of GRIN lens.

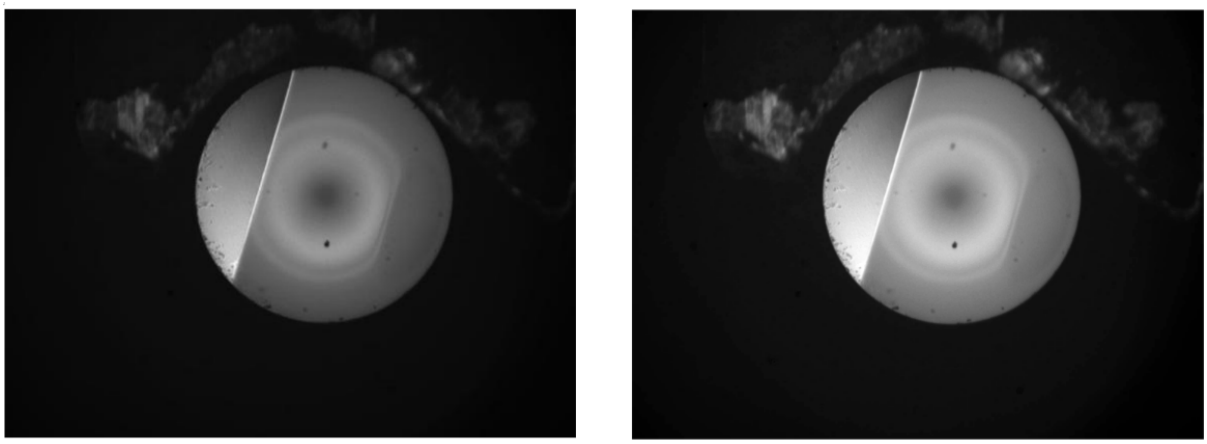

Figure 3. Central brightness enhancement.

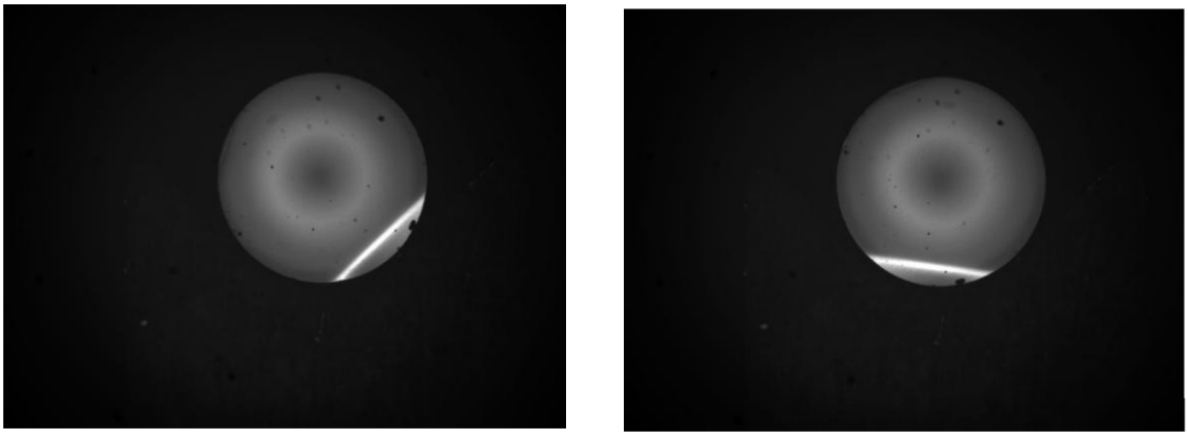

Figure 4. Rotation.

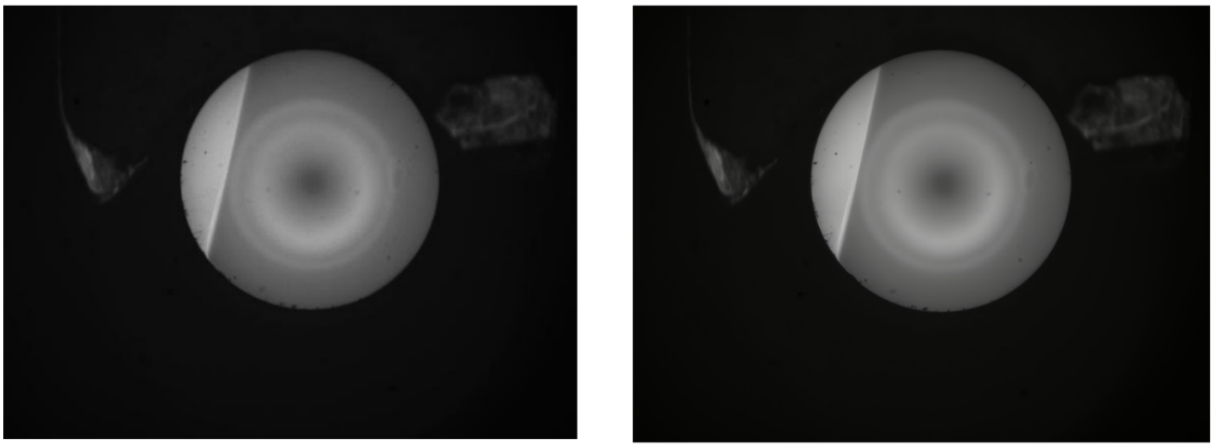

Figure 5. Image quality improvement.

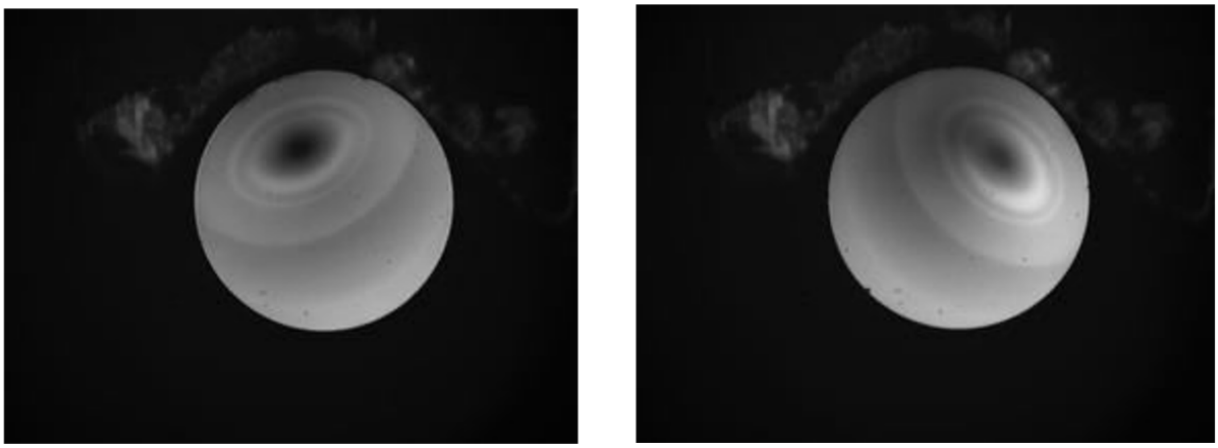

Figure 6. Flipping.

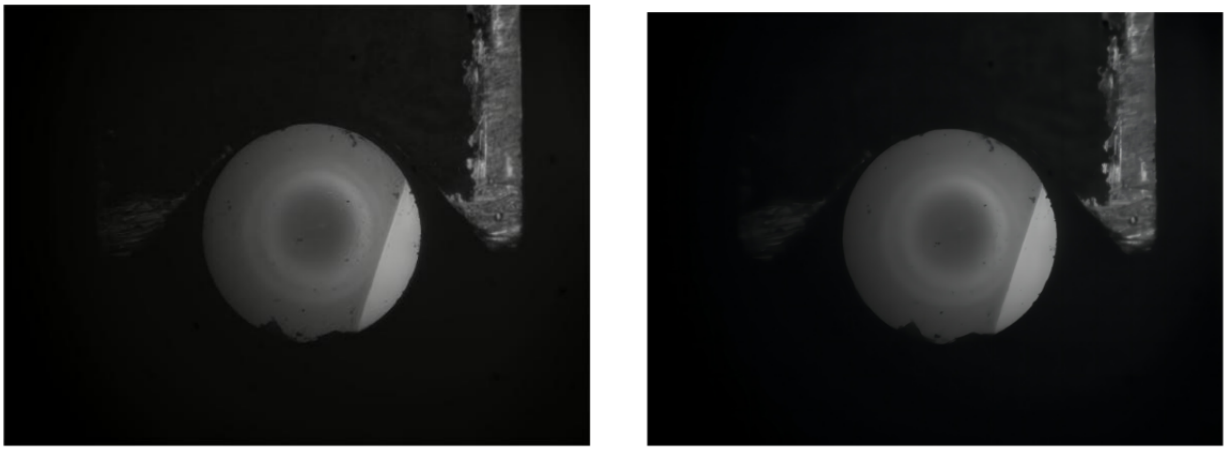

Figure 7. Noise removal.

Table 1. Sample distribution of the dataset

Category	Number
Front	488
Opposite	540

3. Classification network model experiment

3.1. Model selection experiment

The experimental development environment for the classification model selection is the Pytorch2.0.1 deep learning framework, CUDA11.8, and the GPU model is NVIDIA Ge Force RTX 3050 Laptop. As shown in **Table 2**, the dataset divides the lens end face images into a training set and a validation set in an 8:2 ratio. Two models with good performance indicators on both datasets are selected as classification models. The specific operation is to call the weights trained from the Image Net dataset, freeze all layers of the CNN model, set Trainable to False, and only train the classifiers added by oneself. That is, move the learned front layers into the lens end face classification image classification model. The model parameters are shown in **Table 3**. This article sets the number of iterations to 50, with an initial learning rate of 0.001, and combines the cross-entropy loss function with the Adam optimization algorithm.

Table 2. Sample distribution of the dataset

Dataset category	Training set	Validation set
Front	488	122
Opposite	540	135

Table 3. Experimental model parameters

Category	Configuration details
Optimizer	Adam
Initial learning rate	0.001
Loss function	Cross-entropy loss function
Batch-size	32
Epochs	50
Classifier	Dense (1024) + Dropout (0.5) + Flatten + Dense (2) + Softmax

The model selection experiment was conducted twice under the same conditions, and the running time of the model was recorded. The final accuracy and loss value is the average of the accuracy and loss values of two experiments. The loss values Loss1 and Loss2 of the two tests correspond to the accuracy rates Acc1 and Acc2 of the two tests, respectively. **Table 4** to **Table 7** show the comparison results of classification performance between different models on the dataset.

Table 4. Performance comparison of datasets on Res Net

Model	Loss1	Loss2	Acc1	Acc2	Time1	Time2	MLoss	MAcc
Resnet18	0.5338	0.4917	0.7370	0.7143	22m11s	23m14s	0.5128	0.7257
Resnet34	0.3178	0.3416	0.8742	0.8561	21m41s	21m40s	0.3297	0.8651
Resnet50	0.3109	0.2926	0.8884	0.8910	23m41s	23m50s	0.3018	0.8893
Resnet101	0.3123	0.3387	0.8905	0.8653	27m28s	28m7s	0.3225	0.8779

Table 5. Performance comparison of datasets on VGG Net

Model	Loss1	Loss2	Acc1	Acc2	Time1	Time2	MLoss	MAcc
VGG16	0.4175	0.3869	0.7970	0.8101	25m13s	25m18s	0.4022	0.8036
VGG19	0.3966	0.3852	0.8302	0.8261	26m52s	26m58s	0.3909	0.8282

Table 6. Performance comparison of datasets on mobile net

Model	Loss1	Loss2	Acc1	Acc2	Time1	Time2	MLoss	MAcc
	0.4557	0.4689	0.8146	0.7924	31m26s	30m59s	0.4623	0.8035
	0.4750	0.4699	0.7900	0.7954	31m50s	32m22s	0.4725	0.7927

Table 7. Performance Comparison of Datasets on Dense Net

Model	Loss1	Loss2	Acc1	Acc2	Time1	Time2	MLoss	MAcc
DenseNet-121	0.5620	0.5620	0.8073	0.8037	21m34s	20m47s	0.5620	0.8055
DenseNet-169	0.2720	0.2659	0.8965	0.9105	22m42s	22m40s	0.5805	0.8215
DenseNet-201	0.3582	0.3707	0.8462	0.8281	25m41s	23m37s	0.3645	0.8372

3.2. Analysis of experimental results

By comparing the model performance in **Table 2** to **Table 5**, it can be seen that VGG Net and Mobile Net have significant differences in time performance indicators on the dataset, so improvements to these two models are excluded. ResNet50 demonstrates strong overall performance on the GRIN lens end-face image dataset, whereas DenseNet-169 achieves faster classification speed and shorter model runtime, albeit with slightly lower overall performance. The results indicate that the models exhibit strong robustness on this dataset, achieving favorable performance in terms of both loss values and classification accuracy. Overall comparison. This article chooses to use ResNet50 and DenseNet-169 models as the improved basic network models.

4. Attention mechanism module

The attention mechanism originates from human visual research. When dealing with This article chooses to add attention mechanism modules to improve the ResNet-50 and DenseNet-169 models. Common attention

mechanism modules include STN, SE, CBAM, etc [9,10]. In order to select the most suitable attention mechanism module for the dataset in this article, the impact of using SE and CBAM on network performance was analyzed to determine the most suitable attention mechanism module.

4.1. SE module

The implementation of SE-Net begins with applying global average pooling to the input feature map of size ($W \cdot H \cdot C$), thereby compressing spatial information and generating a channel-wise descriptor of size ($1 \cdot 1 \cdot C$) [11]. Subsequently, this descriptor is passed through two fully connected layers. The first fully connected layer reduces the channel dimensionality to ($1 \cdot 1 \cdot C/r$), where is the reduction ratio, while the second layer restores the dimensionality to ($1 \cdot 1 \cdot C$), enabling the model to capture inter-channel dependencies with reduced computational complexity.

The resulting channel weights are then normalized using a softmax function and applied to the original feature map through channel-wise multiplication, producing an output feature map of size ($W \cdot H \cdot C$). Compared to the original input features, the SE module selectively emphasizes informative channels while suppressing less relevant ones, thereby improving feature representation. The overall structure of SE-Net is illustrated in **Figure 8**.

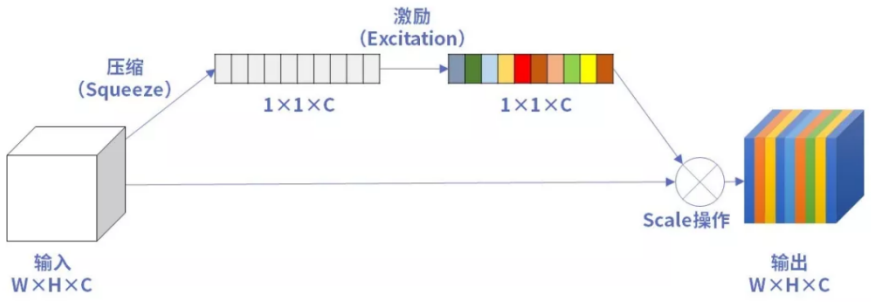

Figure 8. SE net structure diagram.

4.2. CBAM module

The CBAM module consists of two parts: a Channel Attention Module and a Spatial Attention Module. The overall module of CBAM is shown in **Figure 9**.

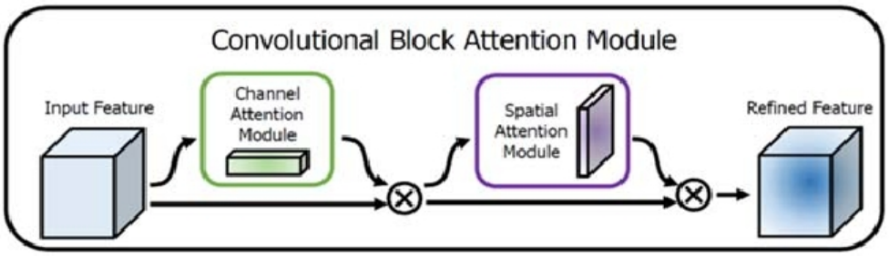

Figure 9. CBAM module structure diagram.

Compared to SE, CAM only has an additional parallel maximum pooling layer operation. Study suggests through experiments that the maximum pooling operation can collect more image feature information and obtain finer channel attention [12]. Therefore, using both average pooling and maximum pooling operations

can simultaneously greatly improve the network's representation ability, which is superior to using a single operation. CAM is shown in **Figure 10**.

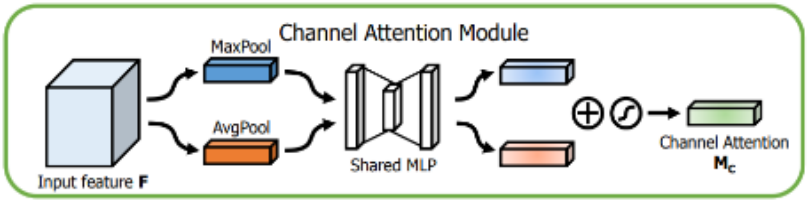

Figure 10. CAM module structure diagram.

The calculation formula for CAM is shown in **Equation 1**:

$$M_c(F) = \sigma\left(MLP\left(Avgpool(F)\right) + MLP\left(Maxpool(F)\right)\right) \quad (1)$$
$$= \sigma\left(W_1\left(W_o\left(F^c_{avg}\right)\right) + W_1\left(W_o\left(F^c_{max}\right)\right)\right)$$

Among them, σ represents an activation function. Firstly, the average pooling and maximum pooling operations are used to aggregate the spatial information of the feature map, generating two different spatial context descriptions F^c_{avg} and F^c_{max}, respectively, the average pooling feature and the maximum pooling feature. Secondly, the generated two features are fed into a shared network consisting of a multi-layer perceptron (MLP) and a hidden layer. Additionally, the generated two features are fed into a shared network consisting of a multi-layer perceptron (MLP) and a hidden layer.

4.3. Experimental comparison and result analysis

There are three main improvements to ResNet50 and DenseNet-169. Firstly, an attention module was added to the network model to improve classification accuracy. Secondly, use the fully connected layer in the attention module with 1×A convolutional layer of size 1 is used to reduce the number of model parameters and reduce model complexity. Finally, the global average pooling layer is used as the classifier. This section conducts comparative experiments on the SE module and CBAM module. The experimental details are shown in **Table 8**.

During training, do not freeze any layers, use dynamic learning rate, set the maximum learning rate to 1e-4, the minimum learning rate to 1e-6, the learning rate scaling ratio to 0.3, set the Patience to 2, monitor the loss of the validation set, and use global flat pooling instead of fully connected layers.

Table 8. Experimental details configuration

Category	Configuration details
Optimizer	Adam
Initial learning rate	1e-4
Batch-size	16
Epochs	50

4.4. Evaluation indicators

This model uses confusion matrix, accuracy, precision, recall, and specificity as evaluation indicators. The confusion matrix is shown in **Table 9**.

Table 9. Confusion matrix

Truth	Forecast results	
	Positive example	**Counterexample**
Positive example	TP（True positive）	FP（False negative）
Counterexample	FN（False positive）	TN（True negative）

Accuracy represents the proportion of correctly classified types of samples in the total sample size of the model, calculated using **Equation 2**.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FN+FP}} \times 100\% \quad (2)$$

Precision represents the actual positive cases are predicted among all positive cases predicted by the model, and is calculated using **Equation (3)**.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \quad (3)$$

Recall rate represents the proportion of all real positive cases that the model correctly predicts as positive cases, calculated using **Equation (4)**.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \quad (4)$$

The specificity represents the proportion of true counterexamples that the model correctly predicts as counterexamples, calculated using **Equation (5)**.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \times 100\% \quad (5)$$

4.5. Analysis of experimental results

Firstly, comparative experiments were conducted on ResNet50 and DenseNet169 before and after adding SE to the dataset, respectively. Before adding the attention mechanism, select the group with the smallest loss value and the highest accuracy rate on the dataset in the previous section for the model weight selection. To facilitate the comparison of experimental results, set the attenuation ratio to 16. In order to facilitate the comparison and detection of model performance indicators, this article calculates the Precision, Recall, and Specificity values for each category. The confusion matrix is shown in **Figure 11**, and the evaluation indicators are shown in **Table 10**.

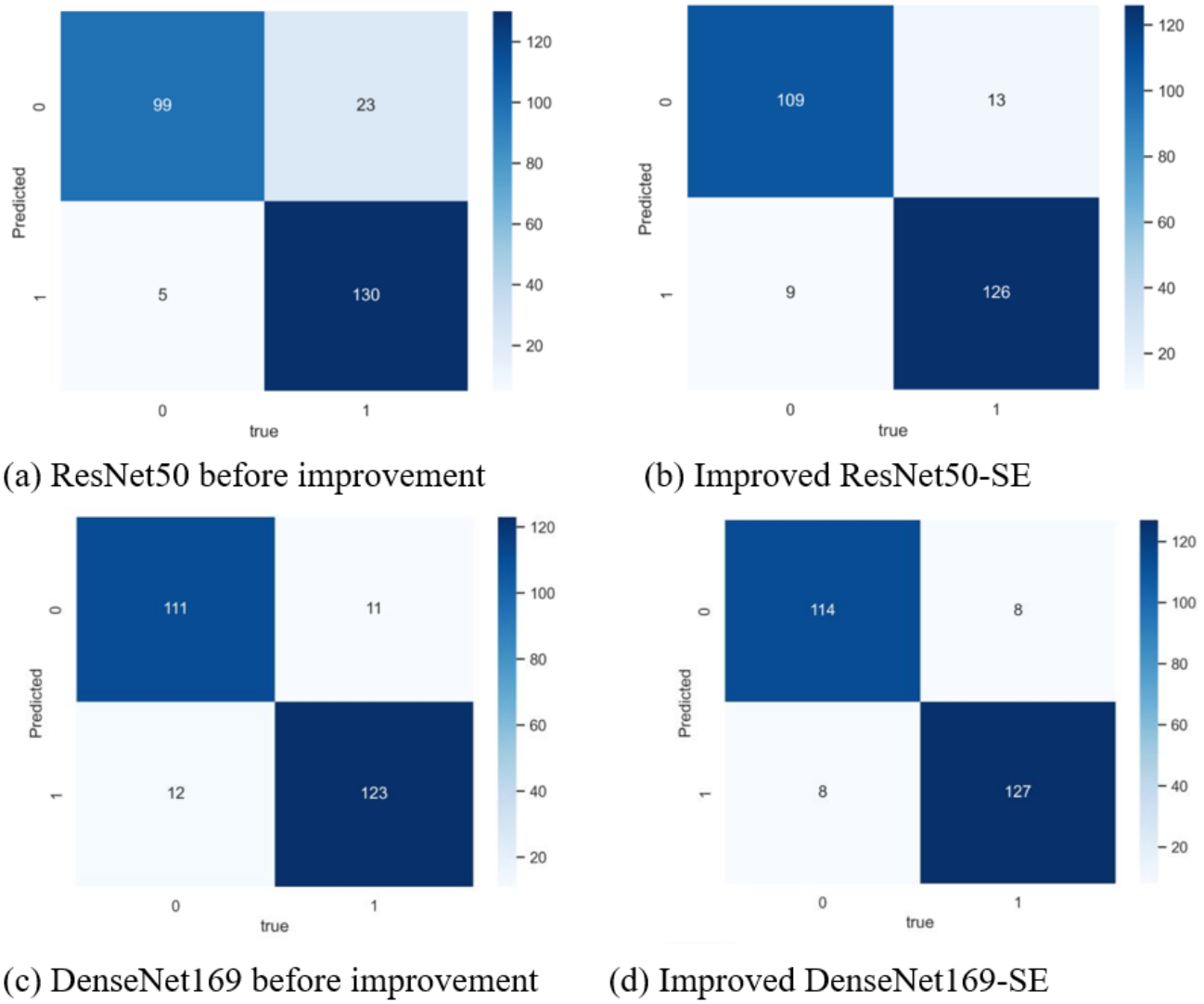

Figure 11. Comparison results of confusion matrix before and after model improvement on ResNet50 and DenseNet169.

Table 10. Comparison of performance indicators for each category on the dataset

Category	Before improvement				Improved			
	Accuracy	Precision	Recall	Specificity	Accuracy	Precision	Recall	Specificity
ResNet50	0.8910	0.8115	0.9519	0.8496	0.9144	0.8934	0.9316	0.9065
DenseNet169	0.9105	0.9098	0.9024	0.9179	0.9377	0.9344	0.9344	0.9407

From **Table 10**, it can be seen that the accuracy of the ResNet50 model on the dataset increased from 0.8901 to 0.9144, an increase of 0.0243. The precision has increased by 0.0819. Recall rate decreased by 0.0203. Specificity increased by 0.0569. The accuracy of the DenseNet169 model on the dataset increased from 0.9105 to 0.9377, an increase of 0.0272. Precision increased by 0.0246. Recall rate increased by 0.032. The specificity increased by 0.0228. After adding the SE module, the performance indicators of ResNet50 and DenseNet169 on the dataset have increased.

Moreover, comparative experiments were conducted on ResNet50 and DenseNet169 before and after adding CBAM modules to the dataset. Before adding the attention mechanism, select the group with the smallest loss value and the highest accuracy rate on the dataset in the previous section for the model weight selection. To facilitate the comparison of experimental results, set the attenuation ratio to 16, and to facilitate the comparison and detection of model performance indicators. This article calculates the Precision, Recall, and Specificity values for each category. The confusion matrix is shown in **Figure 12**, and the evaluation indicators are shown in **Table 10**.

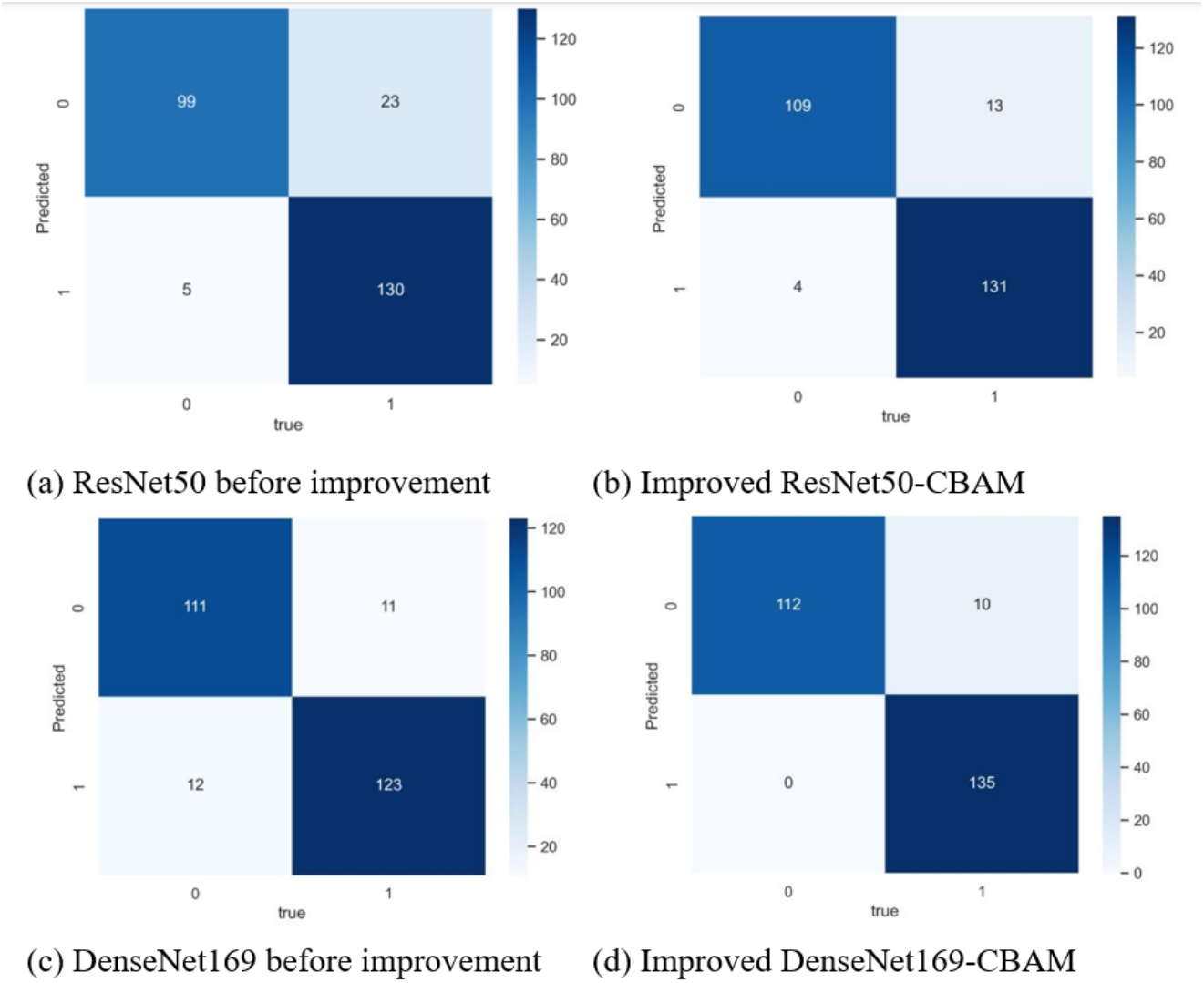

Figure 12. Comparison results of the confusion matrix before and after model improvement on ResNet50 and DenseNet169.

From **Table 11**, it can be seen that the accuracy of the ResNet50 model on the dataset increased from 0.8901 to 0.9338, an increase of 0.0437. Precision increased by 0.0819. Recall rate decreased by 0.0127. The specificity increased by 0.0601. The accuracy of the DenseNet169 model on the dataset increased from 0.9105 to 0.9611, an increase of 0.0506. Precision increased by 0.0082. Recall rate increased by 0.0976. Specificity increased by 0.0131. After adding the CBAM module, the performance indicators of ResNet50 and DenseNet169 on the

dataset have increased.

Table 11. Comparison of performance indicators for each category on the dataset

Category	Before improvement				Improved			
	Accuracy	Precision	Recall	Specificity	Accuracy	Precision	Recall	Specificity
ResNet50	0.8910	0.8115	0.9519	0.8496	0.9338	0.8934	0.9646	0.9097
DenseNet169	0.9105	0.9098	0.9024	0.9179	0.9611	0.9180	1	0.9310

5. Conclusion

To determine the most suitable classification model for this dataset, a transfer learning parameter fine-tuning method was employed. By comparing the performance indicators of four convolutional neural network models through experiments, ResNet50 and DenseNet169, which demonstrated strong performance on the dataset, were selected as the base models for modification. After incorporating the SE attention mechanism, the accuracy of the ResNet50 model increased from 0.8901 to 0.9144, an improvement of 0.0243, while the accuracy of DenseNet169 increased from 0.9105 to 0.9377, an improvement of 0.0272. Similarly, after adding the CBAM attention module, the accuracy of ResNet50 increased from 0.8901 to 0.9338, an improvement of 0.0437, and DenseNet169 increased from 0.9105 to 0.9611, an improvement of 0.0506. The results indicate that the improved models achieve superior detection performance compared to the original classification models. All performance indicators on the dataset were enhanced, effectively improving the classification accuracy of the original network architectures.

Disclosure statement

The author declares no conflict of interest.

References

[1] Xing J, Jiao M, Liu Y, 2015, Application of Self Focusing Lens in all Solid-State Lasers. Journal of Xi'an University of Technology, 31(2): 127–131+124.

[2] Wu S, Zhang M, Zhang B, 2016, Hierarchical Structure Information and Its Expression for Image Features Extraction and Processing of GRIN Lens End, Science And Engineering Research Center, Proceedings of 2016 International Conference on Electrical Engineering and Automation (ICEEA 2016), 643–646.

[3] Wu S, Zhang B, Zhang M, 2017, An Improved Median Filtering Method and its Applications in Features Extraction of GRIN Lens EndImage".

[4] Chu H, 2011, Research on Surface Defect Detection Technology of Optical Components in High Power Laser Devices Based on Machine Vision, thesis, Chongqing University.

[5] Sowers I, 1999, Optical Cleanliness Specifications and Cleanliness Verification, Proceedings of the 44th Annual Meeting of the International Symposium on Optical Science, Engineering, and Instrumentation, 525–530.

[6] Shi W, 2000, Research on High-Precision Cleanliness Detection Methods, thesis, Sichuan University, 9–15.

[7] Liu G, Wang J, Ning R, et al., 2021, Self Focusing Lens End Face Image Processing and Defect Feature Extraction.

Electronic Production, 2021(5): 74–76.

[8] Wu J, 2020, Research and Implementation of Fine-Grained Emotion Classification Method Based on Attention Mechanism, thesis, Heilongjiang University.

[9] Bas A, Huber P, Smith W, et al., 2017, 3D Morphable Models as Spatial Transformer Networks, 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), 895–903.

[10] Woo S, Park J, Lee J, et al., 2018, Cbam: Convolutional Block Attention Module. 2018 European Conference on Computer Vision (ECCV), 3–19.

[11] Jie H, Li S, Samuel A, et al., 2020, Squeeze-and-Excitation Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(8): 2011–2023.

[12] Woo S, Park J, Lee J, et al., 2018, CBAM: Convolutional Block Attention Module, The European Conference on Computer Vision, 3–19.