Journal of Nonlinear and Convex Analysis
Volume 26, Number 7, 2025, 2261–2278

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

A SEMI-SUPERVISED LEARNING IMAGE CLASSIFICATION METHOD FOR MULTISCALE FILTER CONVOLUTIONAL NEURAL NETWORKS

JINBO QIU, DELONG CUI*, ZHIPING PENG, QIRUI LI, JIEGUANG HE, AND JIANBIN XIONG

Abstract. The original semi-supervised learning method has several limitations, such as a small number of labeled samples, high labeling cost, and difficulty in obtaining labels, which restrict the performance and application range of image classification models. Therefore, this study proposed a semi-supervised learning method using multiscale filter convolutional neural network with unlabeled image data to enhance the model performance. A convolutional neural network model was constructed using labeled data for model training and parameter optimization. The trained model was used to predict the unlabeled image data, and the pseudo-labels were generated based on the prediction results. Then, labeled and pseudo-labeled data were amplified and spliced to form a larger training set. The amplified training set was used to retrain the model, and the semi-supervised learning was carried out by combining the labeled and the pseudo-labeled data. The experimental results showed that the proposed method improved the accuracy and robustness of image classification by iterating the process of generating false labels and training models when the number of labeled samples was extremely limited.

1. Introduction

Image classification is a widely used technology in various fields, including facial recognition, medical image analysis, and autonomous driving [30]. The further development of deep learning technology, especially convolutional neural network (CNN), has significantly improved the accuracy of image classification, thus significantly optimizing the performance of image classification tasks. At present, for improving the accuracy of image classification, most supervised learning methods rely on large amounts of labeled data for training. However, data annotation is still predominantly a manual process, making it time-consuming, prone to errors, and ultimately reducing the algorithm's performance.

2020 *Mathematics Subject Classification.* 68T07, 68U10.

Key words and phrases. Data amplification, multiscale filter, residual learning, self-training algorithm, semi-supervised learning, spurious label.

The work presented in this paper was supported by: National Natural Science Foundation of China (62273109); Key Realm R&D Program of Guangdong Province (2021B0707010003); Guangdong Basic and Applied Basic Research Foundation (2022A1515012022, 2023A1515240020, 2023A1515011913, 2024A1515012090); Key Field Special Project of Department of Education of Guangdong Province (2024ZDZX1034); Maoming Science and Technology Project (210429094551175, 2022DZXHT028, mmkj2020033); Projects of PhDs' Start-up Research of GDUPT (2023bsqd2012, 2023bsqd1002, 2023bsqd1013, XJ2022000301).

*Corresponding author.

Accurate labeling of image data in large quantities is not only costly but also time-consuming and labor-intensive, especially in some specialized or rare categories of image data, such as military, industrial, and other confidential environmental data. In 2019, Berthelot et al. proposed the ReMixMatch algorithm [2], which mainly improved the semi-supervised learning performance through distribution alignment and enhanced anchoring. However, its disadvantage was that the recognition performance was unstable when dealing with highly unbalanced dataset. Qizhe Xie et al. proposed the Noisy Student algorithm in 2020 [33]. However, its disadvantage was that when the quality of labeled data was poor, the training performance of the model was reduced. This occurred because of the presence of noisy or incorrect labels in the initial labeled data during the model's self-training stage. These errors were amplified in the algorithm iteration, reducing the overall performance of the classifier. Yassine Ouali et al. proposed the SimPLE algorithm in 2021 [9], which mainly enhanced the performance of semi-supervised learning through similar pseudo-label generation. However, its disadvantage was that it relied on high-quality pseudo-label generation, which, in practical engineering applications, led to a decline in recognition accuracy due to the complexity and diversity of the data.

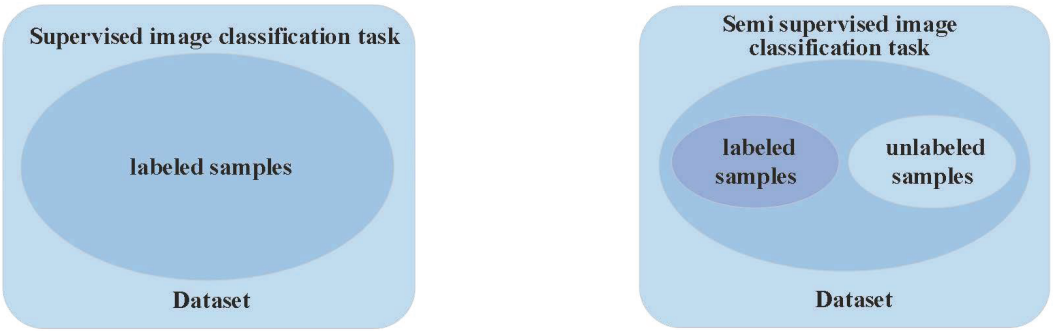
A single semi-supervised method can no longer address the challenge of generating high-quality false labels. Consequently, it fails to improve recognition accuracy, especially in emerging physical layer fields, such as new complex nanocomposites, thermal conductivity of polymeric nano-composites (PNCs), carbon nanotube–reinforced polymer composites [15,18,20,22,22], and so on. Carlos Quintero Gull et al. (2024) proposed an algorithm based on the learning algorithm for multivariate data analysis), which calculated the degree of membership between data and groups/classes [7]. This study defined a membership threshold, enabling the assignment of individuals to classes or clusters with membership values exceeding the threshold. However, a limitation of this approach was that determining the appropriate membership threshold could be problematic due to data scarcity in datasets with low-dimensional data or descriptors. Leixin Qi et al. (2024) proposed a new method combining class-incremental learning and semi-supervised learning to accurately identify loads from large streams of unlabeled data [25]. It also prevented catastrophic forgetting by retaining samples, extracting knowledge, and adjusting the weights of incremental tasks with unlabeled data, addressing semi-supervised problems in incremental learning. Zechen Liang et al. (2024) proposed a semi-supervised learning method combining double-threshold screening and similarity learning. The threshold of another adaptive class was extracted from the labeled data. This class-adaptive threshold could screen many unlabeled data whose predictions were below 0.95 but above the extracted threshold for model training. The method also improved the effectiveness of the overall loss function. However, it was unsuitable for experimental settings with sparse labeled data. This was because the class-adaptive threshold was extracted from the labeled data during model training. When the number of labeled data was too small, the extracted class-adaptive threshold was not representative, leading to the inclusion of many noisy unlabeled data in model training [14].

Therefore, in cases where annotation data are limited, the method to improve the classification performance of image classification still has room for optimization. In

this study, a semi-supervised learning image classification method based on CNNs of multiscale filters was proposed to improve the accuracy of image classification and reduce the dependency on large amounts of labeled data. The generalization performance of the model was enhanced by efficiently using the unlabeled data, thereby maintaining or improving classification accuracy while reducing the need for labeled data.

2. Image classification technique

Image classification tasks are classified into supervised and unsupervised tasks according to whether the training data are labeled or not [34]. Figure 1(a) shows a supervised learning dataset with complete labels. In unsupervised learning, samples are unlabeled and no clear boundary exists between data. The network learns the regularity between data through training to complete the classification task [5]. Figure 1(b) shows the semi-supervised learning dataset, combining supervised and unsupervised learning, with a small number of samples undergoing supervised learning and the remaining samples undergoing unsupervised learning.



(A) (B)

Figure 1. Dataset distribution diagram: (A) supervised image classification tasks and (B) unsupervised image classification tasks.

The traditional supervised image classification method uses manual feature extraction algorithm and statistical classifier to classify images into predefined categories. The main steps include data preprocessing, feature extraction, screening and dimensionalization reduction, classifier training, and performance evaluation. When the algorithm is preprocessed, the classification effect of the image data is improved in steps such as grayscale, normalization, enhancement, and de-noising. SVM, decision trees, and other methods train the classifier, evaluated by accuracy, recall, F1 score, and confusion matrix.

Semi-supervised image classification combines supervised and unsupervised learning using a small number of labeled samples for supervised learning and the remaining unlabeled data for unsupervised learning. This method lowers the dataset requirements and is suitable for tasks where obtaining a high-quality dataset is more difficult. For the image classification tasks, a small amount of annotated data helps the network layer to define data boundaries, improving classification accuracy,

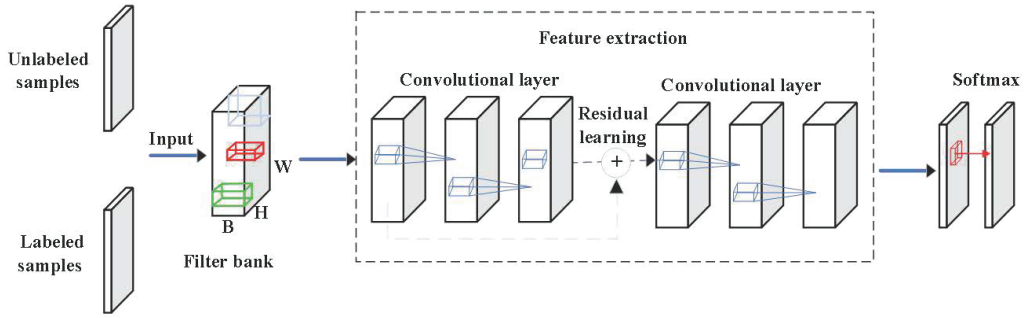


FIGURE 2. Structure diagram of the MF-CNN model

reducing the application standard, and enhancing the flexibility and feasibility of classification tasks [13].

3. Semi-supervised classification algorithm for CNNs of multiscale filters

3.1. **Multiscale filter CNNs.** The limited number of samples in real-world applications often restricts the accuracy of image classification because the feature information obtained from an image is insufficient, so the accuracy of the image classification has great room for improvement [33]. In this study, we proposed an optimization scheme for CNNs by incorporating multiscale filter banks and residual learning modules into traditional CNNs to obtain more image-related feature information. As shown in Figure 2, the multiscale filter CNN (MF-CNN) model is mainly composed of multiscale filter banks, convolutional layers, residual learning modules, Softmax classifiers, and so forth.

In the convolutional layer of the MF-CNN network model, a 5×5 convolution kernel is used to perform convolutional operations. This involves multiplying the convolution kernel with elements of the input, and then summing the results to obtain an output feature [28]. By learning the parameters in the convolution kernel, the network can extract the local image features. The convolution layer adopts multiple matrices to maximize the capture of image content, allowing the extraction of the image information more effectively.

$$Z = g\left(\sum_{i=1}^{n} \omega_i x_i + b\right). \tag{3.1}$$

Where $w_i - w_n$ represents the weight, $x_i - x_n$ represents the component of the input vector, b represents the bias, and $g(x)$ represents the activation function [24]. The filter size is $m \times n$, which can be regarded as a matrix, and its role is to accurately extract feature information from the image. This process involves selecting an area of the image that matches the filter size and performing product and accumulation operations within this area. Each element of the filter is multiplied by the pixel values in the corresponding area of the image, and the product results are then added to arrive at a single value. The matrix moves from the left side of the image to the right based on a specific step size, repeating the product and accumulation

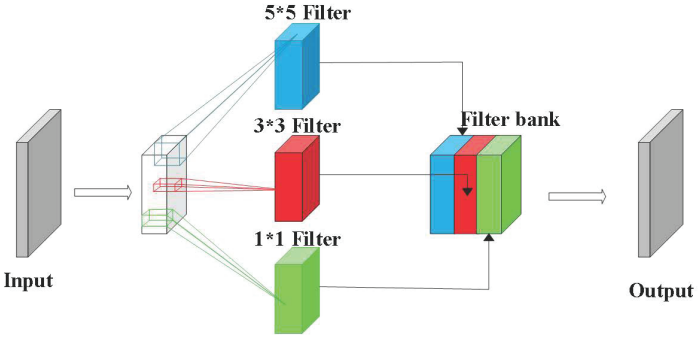


FIGURE 3. Schematic diagram of the multiscale filter banks.

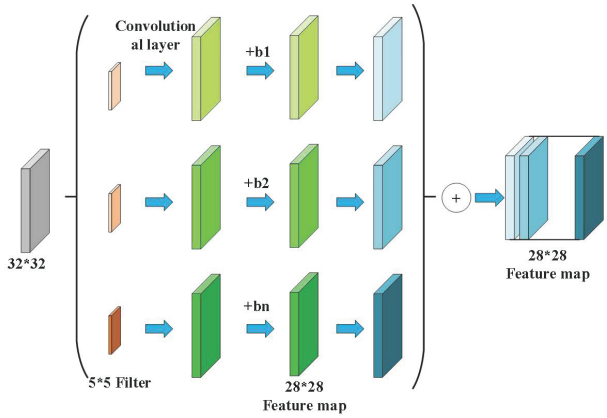


FIGURE 4. Convolutional operation diagram of the MF-CNN model filter.

operations described earlier with each move. As the filter moves, it traverses every region of the image and adds up all the data results. When the filter traverses the entire image, it completes the filtering operation of the entire image. Such product and summation operations between matrices are consistent with the convolution operations in CNN. The MF-CNN model uses a multiscale filter bank, which has three filters of different sizes, and each filter can extract features of different scales and details from the image so as to obtain the feature information of the image from multiple angles. Figure 3 shows the multiscale filter bank of the MF-CNN model.

The feature maps generated by the three filters adopt a unified size, ensuring consistency in size and enabling them to merge into a joint feature map. Zero-padding is applied around the image, with the filter bank producing feature maps of sizes $(H + 4, W + 4)$, $(H + 2, W + 2)$.

where H the matrix height and W is the width. For example, if the MF-CNN input is a 32×32 matrix, three 5×5 filters convolve with it. Through the convolution operation of the filter, the generated image feature information forms a matrix a 28×28 matrix. The specific calculation process is shown in Figure 4.

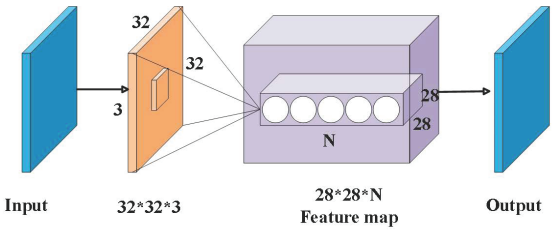


FIGURE 5. Data processing of the convolutional layer in the MF-CCN model.

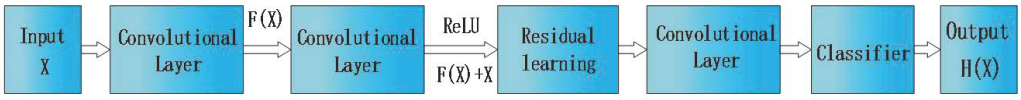


FIGURE 6. Residual learning cross-layer transmission process.

Figure 5 shows the data processing flow in the convolutional layer. Once the feature map passes through the filter, it is passed onto the next layer as input; the number of cores is set to 128, and 128 feature maps are finally obtained, all of which are 28×28 in size.

In multilayer neural network structures, the control of values plays a crucial role in the stability and overall performance of the network. When the gradient exceeds 1 during layer propagation, especially in deep networks, it can grow exponentially, causing gradient explosion. Conversely, when the gradient value is less than 1, as the information is transmitted layer by layer, the gradient gradually decreases, potentially leading to gradient disappearance. The output of the previous layer is directly transmitted to deeper network layers. This way, even in deep networks, the gradient maintains a certain value, which is not too large to cause the gradient to explode, nor too small to cause the gradient to disappear. Through this approach, the neural network in the algorithm better learns the features of the data and improves the performance of the model [1]. Equation (3.2) defines this process:

$$y = F(x, \{W_i\}) + x \tag{3.2}$$

where x is the input of the convolutional layer, and y is the output. The function $F = y - x$ is the residual mapping from the convolutional filter input to the residual output $y - x$. The residual learning process across layers is shown in Figure 6.

The classifier plays a crucial role in image recognition tasks and performs detailed classification operations on each input image. After complex calculations and processing, the classifier generates an output vector that represents the probability of each sample corresponding to each category. The cross-entropy loss function is a commonly used evaluation metric in classification problems, which intuitively reflects the difference between the predicted and true values of the model. As the

difference gradually decreases, the predictive ability of the model constantly improves, achieving more accurate classification results, as shown in equation (3.3):

$$H(p,q) = -\sum_{i=1}^{n} p(x_i)\log(q(x_i)). \tag{3.3}$$

Where $p(x_i)$ is expected probability distribution and $q(x_i)$ is the actual probability distribution [27]. Cross entropy is used to evaluate the difference between two different probability distributions under the same variable. In the classification task, Softmax processes the output results to sum the predicted values of multiple classifications to 1 and then uses cross-entropy to determine the loss value.

If the network output is, $z_1 - z_n$ then formula (3.4) is the formula of Softmax function. Where z_i represents the output value of the first node, c is the number of output nodes, that is i, the number of categories of the classification, e represents the constant.

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{c=1}^{c} e^{z_c}}. \tag{3.4}$$

3.2. **Implementation of self-training algorithms.** This study used a self-training algorithm to effectively reduce the cost of image labeling using unlabeled image data. During the algorithm operation, the confidence level of each category was calculated and used as a basis to filter the unlabeled data predicted by the model, as shown in equation (3.5).

$$\text{Confidence}(\mathrm{x}_l) = \text{Rank}_{\mathrm{T}}(\mathrm{p}(y|x_i)). \tag{3.5}$$

Where x and y are used to represent the unlabeled images and the labels predicted by the model for these images, respectively, and $p(y|x_i)$ represent the posterior probability for each image. In addition, *Rank* represents the ranking of posterior probabilities for all images in each category, while T serves as the threshold for ranking, for example, 10% represents the top 10% of posterior probabilities.

The workflow of the self-training algorithm was as follows: First, the SSL-CNN model was trained using labeled samples. Second, the trained classification model was used to infer unlabeled data and generate corresponding pseudo-labels for these unlabeled samples. Third, the generated pseudo-labels were filtered based on the confidence level of the image, retaining only those image data with higher confidence levels and their corresponding pseudo-labels. Fourth, the filtered image data were added to both the labeled and unlabeled datasets for subsequent model training.

Compared with other supervised learning and semi-supervised training methods, the main advantage of self-training was the efficient use of unlabeled image samples, which improved the robustness and generalization ability of CNN models. The training process was simple and easy to implement [10]. Although self-training had many advantages, identifying samples with high prediction probability was difficult and easily misjudged. In the iterative training of the model, it caused "error accumulation" of erroneous samples [11]. The principle of the self-training algorithm is shown in equation (3.6). The Error_rate was used to quantify the error impact of erroneous samples during a single training process of the model.

$$Error\_accum = 1 - (1 - Error\_rate)^n \tag{3.6}$$

where n represents the number of training iterations, and $Error\_accum$ represents the accumulated error after n iterations.

When the value of $Error\_rate$ is 0.1, the error caused by incorrect samples reaches 10% in each iteration. In this scenario, after five consecutive iterations of training, the accumulated error $Error\_rate$ significantly increases to 0.41 iterations, and the cumulative error still increases to 0.005. This indicates that during the model training process, errors caused by incorrect samples gradually accumulate as the training progresses, further affecting the performance of the model. This study minimized the error caused by incorrect samples by selecting confidence levels.

Amin Golzari Oskouei et al. (2024) proposed a semi-supervised fuzzy C-means method based on feature weights and cluster weights learning [6]. However, using the auxiliary information from class labels to guide clustering still presented unresolved issues. In most semi-supervised fuzzy clustering methods, feature weights and cluster weight learning were often neglected, destroying the clustering algorithm's ability to form the optimal cluster structure. Qi Wei et al. (2024) proposed a method called meta-threshold, which learned a dynamic confidence threshold for each unlabeled instance and required no additional hyperparameters beyond the learning rate [31]. However, this method considered only the sample-level threshold of the pseudo-notation method in semi-supervised learning. It provided not only a solution to reduce the training complexity but also did not really improve the recognition accuracy. Xu Chen et al. (2024) proposed a new unified framework called robust structure-aware semi-supervised learning [4]. This method applied joint semi-supervised virus-reduction robust estimation and network sparse regularization iterations on the Laplace matrix to address the challenging problem of distributed robust semi-supervised learning. However, this method had some limitations in generalization performance, and its advantages in imaging applications were not obvious. Suxia Zhu et al. (2024) proposed a method for selecting federal semi-supervised classification samples based on the changes in the predicted distribution of different data augmentation amounts [36]. The focus of this approach was to improve model performance with fewer involved customers during training, while also increasing the reliability of sample selection. However, this method had advantages primarily in scenarios involving Non-independent uniformly distributed data and a limited number of participating clients, with poor generalization performance. Compared with the aforementioned fusion native methods, this study combined the advantages of deep learning and had certain benefits in generalization ability and clustering performance across different scenarios.

4. Experiment and result analysis

This study used six classic computer vision dataset: MNIST, Fashion MINIST, CIFAR-10, CIFAR-100, SVHN, and Cat-Dog, which represented image classification tasks from simple to complex, from single to multiple categories, from grayscale to color images, and so forth. The MNIST dataset is a commonly used entry-level dataset containing grayscale images of handwritten digits, suitable for simple number recognition tasks [12]. The Fashion-MNIST dataset is similar to MNIST and contains images of fashion items [29]. The CIFAR-10 dataset is a more challenging

multiclass object recognition task that includes color object images from 10 categories [3]. CIFAR-100 is an extended version of CIFAR-10, which adds more categories and fine-grained classifications, placing higher demands on the recognition ability of the model [35]. The SVHN dataset contains house number images from Google Street View, which is challenging for digit recognition and object recognition in complex backgrounds [13]. The Cat-Dog dataset is an image classification task used to distinguish cats and dogs [26]. The selection of these datasets aims to comprehensively evaluate the generalization ability and performance of the proposed method in different scenarios.

4.1. **Experimental environment and platform setup.** All experimental models in this study were carefully developed and designed on Linux systems to ensure system stability and compatibility. In terms of deep learning models, TensorFlow, an industry-leading deep learning framework, was adopted as the technical support, and the programming language used was Python 3.7. The CPU used was Intel Core i7-7820X with 64 GB memory, and the GPU used was GeForce GTX 2080Ti. CUDA 10.0 and CUDNN were specially configured to fully leverage the advantages of the GPU in deep learning model training.

4.2. **Analysis of experimental results.** As shown in Tables 1 – 6, the MF-CNN model achieved the best classification performance on the MNIST and CIFAR-10 datasets, with values of 99.20% and 82.23%, respectively. In the MNIST dataset, compared with other image classification algorithms SAE, DBN, MLP, MLELM, and HELM, MF-CNN improved the image classification accuracy by 0.6%, 0.33%, 1.81%, 0.16%, and 0.07%, respectively. In the CIFAR-10 dataset, compared with other image classification algorithms such as Tiled CNN, Improved LCC, KDES-A, and PCA Net Network, MF-CNN improved the image classification accuracy by 9.13%, 7.73%, 6.23%, and 5.09%, respectively. Based on the aforementioned analysis, it was concluded that using the network model proposed in this study for image classification could effectively improve the accuracy of image classification.

TABLE 1. Accuracy of the MF-CNN network model on the MNIST dataset

Model	Accuracy rate (%)
SAE	98.60
DBN	98.87
MLP	97.39
MLELM	99.04
HELM	99.13
MF-CNN	**91.34**

Six datasets such as MNIST, Fashion MINIST, CIFAR-10, CIFAR-100, SVHN, and Cat-Dog were used to train the MF-CNN network model. The accuracy and loss function curves were drawn using the classification results of the test images. Meanwhile, the confusion matrix for image classification was also generated. The diagonal of the confusion matrix represented the number of models whose predictions

TABLE 2. Accuracy of the MF-CNN network model on the Fashion-MNIST dataset

Model	Accuracy rate (%)
Random Forest	81.60
Logistic Circuit	87.62
Gradient Boosting Classifier	88.03
4-NearestNeighbour	85.40
MF-CNN	**91.34**

TABLE 3. Accuracy of the MF-CNN network model on the Fashion-MNIST dataset

Model	Accuracy rate (%)
Tiled CNN	73.10
Improved LCC	74.50
KDES-A	76.00
PCA Net Network	77.14
MF-CNN	**82.23**

TABLE 4. Accuracy of the MF-CNN network model on the CIFAR-100 dataset

Model	Accuracy rate (%)
Spatial Pyramids	54.23
Stochastic Pooling	57.49
Maxout Networks	61.43
Network in Network	64.32
MF-CNN	**52.14**

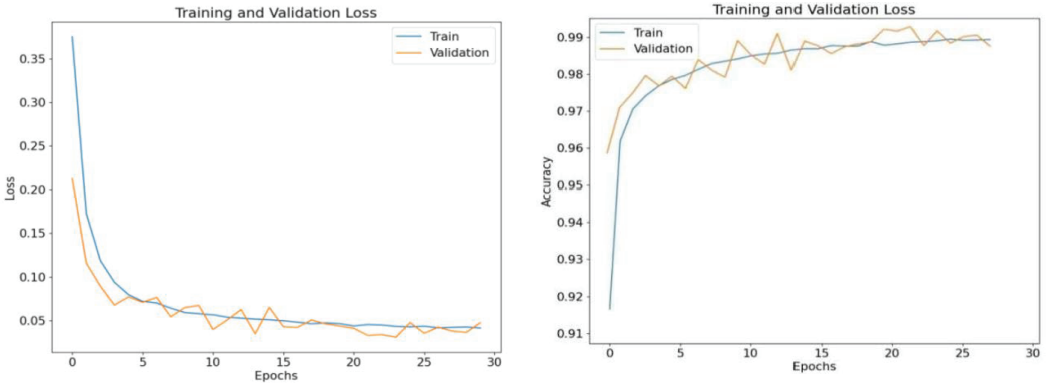
TABLE 5. Accuracy of the MF-CNN network model on the SVHN dataset

Model	Accuracy rate (%)
Spatial Pyramids	63.98
Stochastic Pooling	82.30
Maxout Networks	75.37
Network in Network	85.52
MF-CNN	**85.40**

were consistent with data labels. The larger the value and the better, indicating higher prediction accuracy of the model for this specific particular category.

TABLE 6. Accuracy of the MF-CNN network model on the Cat-Dog dataset

Model	Accuracy rate (%)
Full fine-tuning	82.20
Residual adapters dim = 512	82.28
muNet	92.98
MF-CNN	**87.42**

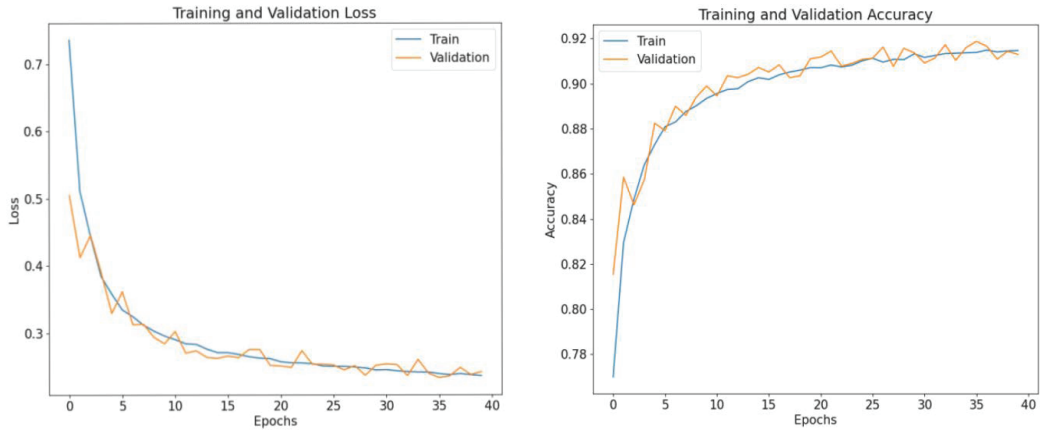


(A) (B)

FIGURE 7. Classification accuracy and loss function value curves of the MF-CNN network model on the MNIST dataset.

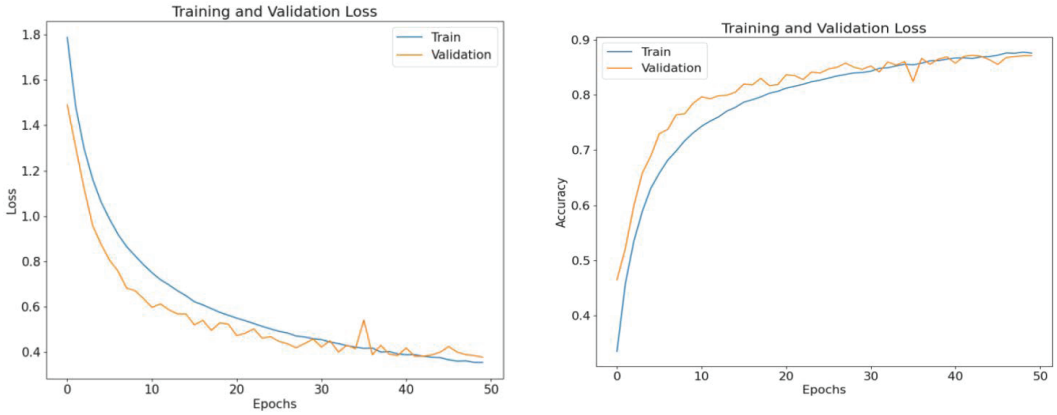
This is because the multiscale filter bank of the MF-CNN model preserved image feature information to the greatest extent, avoiding excessive resource iterations, while residual learning effectively managed gradient issues with feature information. Accuracy and loss value are key indicators, with accuracy measuring performance in recognition and classification. The loss value accurately describes the gap between the predicted results of the model and the actual label value [1]. By comprehensively considering both indicators, we evaluated the model's performance more comprehensively and accurately, offering strong data support for subsequent optimization and improvement [8], as shown in Figures 7-12.

As shown in Figure 8, the classification accuracy and loss function curves of the MF-CNN model were comprehensively analyzed using the MNIST and CIFAR-10 datasets as illustrative examples. During MNIST training, the model achieved 99.03% accuracy on the training set by the 20th iteration and then gradually stabilized effectively. In the first 20 iterations, the accuracy increased rapidly to 99.03%, after which the growth slowed significantly, though it continued to improve slightly. In the 0–20 iterations of the MF-CNN model, the accuracy rate increased rapidly to 99.03%, and then, although it continued to rise, the rate of increase slowed down significantly. Meanwhile, the change trend of the loss function value was the opposite; it declined rapidly at first and then gradually slowed down. In 20–30 iterations,



(A) (B)

FIGURE 8. Classification accuracy and loss function value curves of the MF-CNN network model on the Fashion-MNIST dataset.

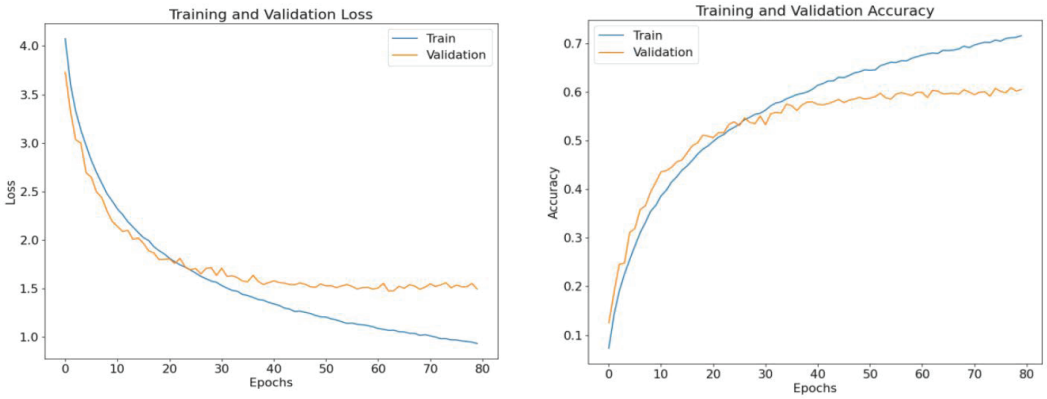


(A) (B)

FIGURE 9. Classification accuracy and loss function value curves of the MF-CNN network model on the CIFAR-10 dataset.

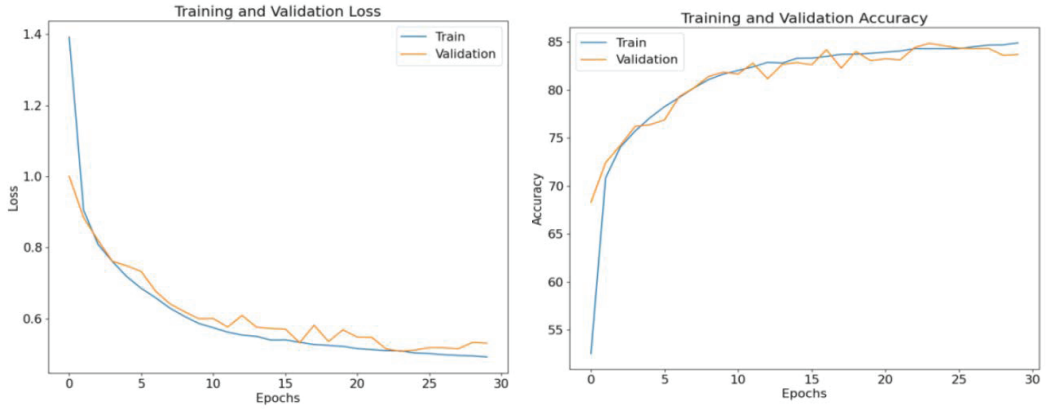
the accuracy and loss values changed very little. After 30 iterations of training, the MF-CNN network model showed excellent performance, with an accuracy of 99.20% and a reduced loss value of 0.023.

On the CIFAR-10 dataset, when the network was trained through 30 iterations, the accuracy of the training set reached 80% and then remained relatively stable over time. Between 30 and 40 iterations, the accuracy rate gradually increased, whereas the loss value slowly decreased, as shown in the Figure 9 below over time. By the 40th iteration, the accuracy reached 80.39%, and the loss value was 0.73. In the 40–50 iteration range, the accuracy and loss values fluctuated relatively little. After 50 iterations, the accuracy of the MF-CNN model further improved to 82.23%, with a reduced loss value of 0.59.



(A) (B)

FIGURE 10. Classification accuracy and loss function value curves of the MF-CNN network model on the CIFAR-100 dataset.

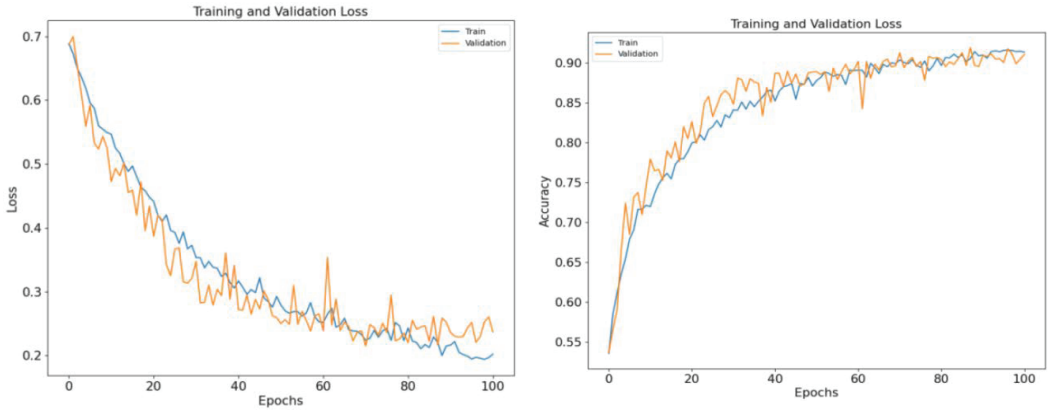


(A) (B)

FIGURE 11. Classification accuracy and loss function value curves of the MF-CNN network model on the SVHN dataset.

These results showed that as training progressed, the accuracy and robustness of the model continued to improve, whereas the loss value continued to decrease, indicating that the model's learning and generalization abilities continuously improved, along with its predictive accuracy. The same performance was also displayed on CIFAR-100, svhn, and Cat-Dog datasets, as shown in Figures 10–12.

The confusion matrix diagrams in Figures 13 and 14 show the excellent performance of the MF-CNN model on the classification tasks using the MNIST and CIFAR-10 datasets. Confusion commonly evaluate classification. Providing a comprehensive insight into classifier performance by showing the statistical relationships between actual categories and model predictions [32].



(A) (B)

FIGURE 12. Classification accuracy and loss function value curves of the MF-CNN network model on the Cat-Dog dataset.

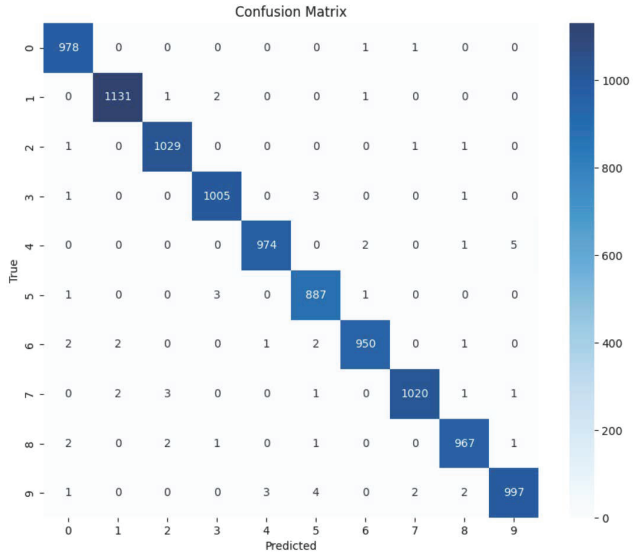


FIGURE 13. Confusion matrix of the MF-CNN model on the MNIST dataset.

Figure 13 shows that the handwritten digit "9" in the single classification results of the MNIST dataset had the lowest accuracy (only 98.81%), whereas the handwritten digit "2" had the highest accuracy, accounting for 99.70%. A total of five samples of the number "4" were classified as number "9"; the number "4" relative to the number "9" had a confusion rate of 0.5%, and the number "9" classified as the number "5" had a confusion rate of 0.3%. Under the MNIST dataset image classification, the final test set accuracy of the MF-CNN network was 99.20%.

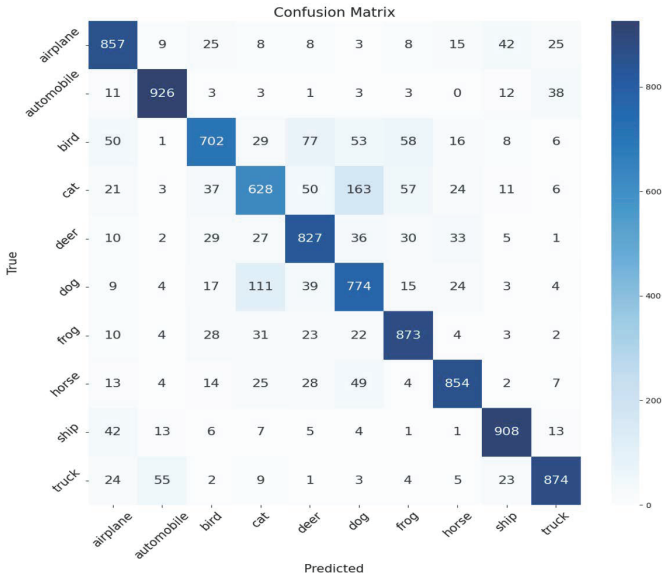


FIGURE 14. Confusion matrix of the MF-CNN model on the CIFAR-10 dataset.

Figure 14 shows that among the single classification results of the CIFAR-10 dataset, the "automobile" category performed the best, with a classification accuracy of 92.60%. In contrast, the "cat" category had the lowest classification accuracy of 62.80%. In the CIFAR-10 image classification task, the "automobile"features were distinct, allowing the neural network to easily identify them.

While the "cat" features were unclear and more difficult to distinguish. A total of 163 "cat" samples were incorrectly classified as "dog", and 111 "dog" samples were incorrectly classified as "cat". These cases of misclassification indicated a certain degree of similarity between "cat" and "dog", which may be reflected in their color, texture, shape, and other aspects, and these common characteristics led to the misjudgment of the model.

The model had a high classification accuracy among different categories. The classification results of each category were distributed near the diagonal of the corresponding category label, indicating that the model successfully classified most images correctly into their respective categories.

5. Conclusions and discussion

The image classification method proposed in this study combined deep learning and semi-supervised learning technologies to improve the performance of the image classification model in the case of scarce annotation data. In the experiment, a series of classical image classification datasets were used to verify and compare with other image classification methods. The results showed that the proposed method could maintain high classification accuracy while reducing dependency on the labeled data, verifying its effectiveness and feasibility in addressing real-world image classification problems. The method proposed in this study can be more suitable for

generalized semi-supervised learning scenarios and has higher recognition accuracy than traditional semi-supervised load identification methods for the following reasons: (1) The multiscale filter convolutional network of deep learning is used to obtain image feature information from multiple angles, addressing the issue of limited storage space by maximizing the image feature information generated by the model. (2) The residual learning method is introduced, which not only retains the model's classification ability for a small number of samples but also addresses the problem of data feature loss caused by the increase or disappearance of gradient values in multilayer neural network structures. (3) The adaptability of semi-supervised load recognition under scene generalization is realized through the organic integration of semi-supervised learning architecture and deep learning methods. Although this study performs well on classical image datasets, we have not yet proposed experimental methods targeted at some special application scenarios to address the issue. Also, the performance may decline when handling more complex datasets, such as the prediction of macroscopic thermal conductivity of polymer graphene-reinforced composites, thermal conductivity of clay-reinforced PNCs, and so forth [16, 17, 19]. Therefore, in future studies, we aim to propose a combination of other artificial intelligence methods, introduce multiscale models at different levels, and use physical information neural networks to conduct experiments [23]. Future experiments will be conducted on real-world datasets to verify the validity and generalization of the method.

References

[1] H. Alaeddine and M. Jihene, *Deep residual network in network*, Computational Intelligence and Neuroscience **2021** (2021): 6659083.

[2] D. Berthelot, N. Carlini, E.D. Cubuk, A. Kurakin, K. Sohn, H. Zhang and C. Raffel, *ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring*, ArXiv, volume=abs/1911.09785, 2019,

[3] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang and Y. Miao, *A review of image classification algorithms based on convolutional neural networks*, Computer Application **42** (2022), 1044–1049.

[4] X. Chen, *Robust structure-aware graph-based semi-supervised learning: Batch ana recursive processing*, ACM Transactions on lntelligent Systems and Technology **15** (2024): article No. 66.

[5] Y. Chi, J. Li and H. Fan, *Pyramid-attention based multi-scale feature fusion network for multispectral pan-sharpening*, Applied Intelligence **52** (2022), 5353–5365.

[6] A. Golzari Oskouei, N. Samadi and J. Tanha, *Feature-weight and cluster-weight learning in fuzzy c-means method for semi-supervised clustering*, Applied Soft Computing Journal **161** (2024): 111712.

[7] C. Q. Gull and J. Aguilar, *A semi-supervised learning algorithm for multi-label classification and multi-assignment clustering problems based on a multivariate data analysis*, Engineering Applications of Artificial Intelligence **137** (2024): 109189.

[8] C. Hu, X. Sun, Z. Yuan and Y. Wu, *Classification of breast cancer histopathological image with deep residual learning*, International Journal of Imaging Systems and Technology **31** (2021), 1583–1594.

[9] Z. Hu, Z Yang, X Hu and R Nevatia, *SimPLE: Similar pseudo label exploitation for semi-supervised classification*, in: Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021, pp. 15094–15103.

[10] Y. Huang and X. Zhong, *RPDNet: Automatic fabric defect detection based on a convolutional neural network and repeated pattern analysis*, Sensors **22** (2022): 6226.

[11] K. Hyung-Soo, S. Jaehwan and J. Hyung-Jo, *Optimal domain adaptive object detection with self-training and adversarial-based approach for construction site monitoring*, Automation in Construction **158** (2024): 105244.

[12] M. S. Ibrahim, A. Vahdat, M. Ranjbar and W. G. Macready, *Semi-supervised semantic image segmentation with self-correcting networks*, in: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 12712–12722.

[13] K. Li and W. Ye, *Semi-supervised node classification via graph learning convolutional neural network*, Applied Intelligence **52** (2022), 12724–12736.

[14] Z. Liang, Y. Wang, W. Lu and X. Cao, *Boosting semi-supervised learning with dual-threshold screening and similarity learning*, ACM Transactions on Multimedia Computing, Communications and Applications **20** (2024): article no. 287.

[15] B. Liu and W. Lu, *Surrogate models in machine learning for computational stochastic multiscale modeling in composite materials design*, International Journal of Hydromechatronics **5** (2022), 336–365.

[16] B. Liu, W. Lu and T. Olofsson, *Stochastic interpretable machine learning based multiscale modeling in thermal conductivity of Polymeric graphene-enhanced composites*, Composite Structures **327** (2024): 117601.

[17] B. Liu, S. R. Penaka and W. Lu, *Data-driven quantitative analysis of an integrated open digital ecosystems platform for user-centric energy retrofits: A case study in Northern Sweden*, Technology in Society **75** (2023): 102347.

[18] B. Liu and N. Vu-Bac, *A stochastic multiscale method for the prediction of the thermal conductivity of polymer nanocomposites through hybrid machine learning algorithms*, Composite Structures **273** (2021): 114269.

[19] B. Liu, N. Vu-Bac and X. Zhuang, *Stochastic multiscale modeling of heat conductivity of Polymeric clay nanocomposites*, Mechanics of Materials **142** (2020): 103280

[20] B. Liu, N. Vu-Bac and X. Zhuang, *Stochastic integrated machine learning based multiscale approach for the prediction of the thermal conductivity in carbon nanotube reinforced polymeric composites*, Composites Science and Technology **224** (2022): 109425.

[21] B. Liu, N. Vu-Bac and X. Zhuang, *Stochastic full-range multiscale modeling of thermal conductivity of Polymeric carbon nanotubes composites: A machine learning approach*, Composite Structures **289** (2022): 115393.

[22] B. Liu, N. Vu-Bac and X. Zhuang, *Al-DeMat: A web-based expert system platform for computationally expensive models in materials design*, Advances in Engineering Software **176** (2023): 103398 .

[23] B. Liu, Y. Wang and T. Rabczuk, *Multi-scale modeling in thermal conductivity of Polyurethane incorporated with phase change materials using physics-informed neural networks*, Renewable Energy **220** (2024): 119565.

[24] T. Liu, J. Chen and X. Li, *Research on image classification based on convolutional neural network*, in: 2022 International Conference on Automation, Robotics and Computer Engineering (ICARCE), IEEE, 2022, pp. 1–4

[25] L. Qiu, T. Yu and C. Lan, *A semi-supervised load identification method with class incremental learning*, Engineering Applications of Artificial Intelligence **131** (2024): 107768.

[26] L. Tang, *Image classification based on improved VGG network*, in: 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP), IEEE, 2021, pp. 316–320.

[27] K. Tripathi, A. K. Gupta and R. G. Vyas, *Deep residual learning for image classification using cross validation*, International Journal of Innovative Technology and Exploring Engineering **9** (2020), 2278–3075.

[28] V. M. Vargas, P.A. Gutiérrez, J. Barbero-Gómez and C. Hervás-Martínez, *Activation functions for convolutional neural networks: Proposals and experimental study*, In: IEEE Transactions on Neural Networks and Learning Systems **34** (2023), 1478–1488.

[29] G. Wan and L. Yao, *LMFRNet: A lightweight convolutional neural network model for omage analysis*, Electronics **13** (2024): 129.

[30] X. Wang, *Research on the application of object detection algorithm based on neural network in the regulation and control of intelligent building energy-saving equipment*, in: Proceedings

of the 2024 3rd International Conference on Frontiers of Artificial Intelligence and Machine Learnin. Association for Computing Machinery, 2024. pp.185–189

[31] Q. Wei, L. Feng, H. Sun, R. Wang, R. He and Y. Yin, *Learning sample aware threshold for semi supervised learning*, Machine Learning **113** (2024), 5423-5445.

[32] Q. Xie, E. H. Hovy, M. Luong and Q. V. Le, *Self-training with noisy student improves ImageNet classification*, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 10684–10695.

[33] C. Yu, Q. Zhang and X. Wu, *Research on the improved graph convolutional neural network algorithm based on historical data analysis in the task of recommending attractions on online travel platforms*, in: 2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering Journal of Anhui University of Technology, IEEE, 2024, pp. 879–884

[34] Y. Yu. *Deep learning approaches for image classification*, in: Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering. Association for Computing Machinery, 2023, pp. 1494–1498.

[35] Y. Zheng, H. Huang and J. Chen, *Comparative analysis of various models for image classification on Cifar-100 dataset*, Journal of Physics: Conference Series **2711** (2024): 012015.

[36] S. Zhu, X. Ma and G. Sun, *Two-stage sampling with predicted distribution changes in federated semi-supervised learning*, Knowledge-Based Systems **295** (2024): 111822.

Manuscript received October 19, 2024
revised February 19, 2025

J. B. Qiu
Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Maoming, China;
College of Electronic Information Engineer, Guangdong University of Petrochemical Technology, Maoming, China
E-mail address: jinboqiu1982@163.com

D. L. Cui
Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Maoming, China;
College of Electronic Information Engineer, Guangdong University of Petrochemical Technology, Maoming, China
E-mail address: delongcui@gdupt.edu.cn

Z. P. Peng
Jiangmen Polytechnic, Jiangmen, China
E-mail address: zhipingpeng@gdupt.edu.cn

Q. R. Li
Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Maoming, China;
College of Electronic Information Engineer, Guangdong University of Petrochemical Technology, Maoming, China
E-mail address: liqirui@gdupt.edu.cn

J. G. He
College of Electronic Information Engineer, Guangdong University of Petrochemical Technology, Maoming, China
E-mail address: jieguanghe@gdupt.edu.cn

J. B. Xiong
School of Automation, Guangdong Polytechnic Normal University, Guangzhou, China
E-mail address: 276158903@qq.com