Journal of Imaging Science and Technology® 68(3): 1–10, 2024.
© Society for Imaging Science and Technology 2024

Research on Transfer Learning from Large-scale Dataset to Few-Shot Image with Similar Background

Quanyou Zhang
State Key Laboratory of Geo-Information Engineering and Key Laboratory of Surveying and Mapping Science and Geospatial Information Technology of MNR, CASM, Beijing, 100036, China
Xuchang University, Xuchang 461000, China
College of Computer Science, Chongqing University, Chongqing 401331, China

Yong Feng
College of Computer Science, Chongqing University, Chongqing 401331, China
E-mail: fengyong@cqu.edu.cn

A-gen Qiu
State Key Laboratory of Geo-Information Engineering and Key Laboratory of Surveying and Mapping Science and Geospatial Information Technology of MNR, CASM, Beijing, 100036, China
E-mail: qiuag@casm.ac.cn

Meng Yin
Chongqing Medical Data Information Technology Co., Ltd, Chongqing 401336, China

Jinling Shi and Yaohui Li
Xuchang University, Xuchang 461000, China

Fangtao Qin
College of Computer Science, Chongqing University, Chongqing 401331, China

Abstract. *Obtaining a large number of unqualified product samples in industrial production is an arduous task. It is challenging to learn the features of few-shot object images. Despite the limited number of original images, we developed a transfer learning method called LDFISB (Large-scale Dataset to Few-Shot Image with Similar Background) that provides a feasible solution. LDFISB is trained on a large-scale dataset such as CIFAR100, and then the model is fine-tuned based on the original model and parameters to achieve classification tasks on a new APSD (auto part surface dataset). Batch normalization, padding, and Weighted Cross Entropy Loss are employed in the training processes. Hyper-parameters are configured according to Hyper-table to enhance the accuracy of the prediction. The CIFAR10, CIFAR100, and ImageNet were considered as pre-training datasets, and the LDFISB method is capable of accurately predicting the flaw area of the product image. The LDFISB method achieves the highest accuracy on the CIFAR100 pre-training dataset.* © *2024 Society for Imaging Science and Technology.*

[DOI: 10.2352/J.ImagingSci.Technol.2024.68.3.030401]

1. INTRODUCTION

Deep learning is a method that automatically learns features from data. It typically requires a large amount of data to train the model to complete the task. However, in a specific environment, it is difficult to obtain a large number of training data, such as focus images of special medical cases and special original pictures in industrial production. Directly using the model to train will produce unsatisfactory results. It is a challenge to apply deep learning to detect the classification of product surface images [1]. For example, there are some sections of defective auto part as shown in Figure 1, which requires a large number of defective images for training a deep learning model before detection. It is difficult to obtain a large number of unqualified product samples in industrial production. To solve this problem, transfer learning [2] provides us with a feasible solution. Although the training dataset is small, classification tasks of the deep learning model can be achieved by fine-tuning. Accomplishing this task is essentially a large model and a large dataset. First, a classification model is trained on a large-scale dataset such as CIFAR100, and then the model is fine-tuned based on the original model and parameters to achieve new image classification.

One notable study in the field of transfer learning is “Deep Residual Learning for Image Recognition” [3], published in 2016. The authors introduced the ResNet architecture [4], which improved upon existing deep neural networks by introducing residual connections between

Received Oct. 9, 2023; accepted for publication Dec. 15, 2023; published online Jan. 19, 2024. Associate Editor: Chao Liu.
1062-3701/2024/68(3)/030401/10/$25.00

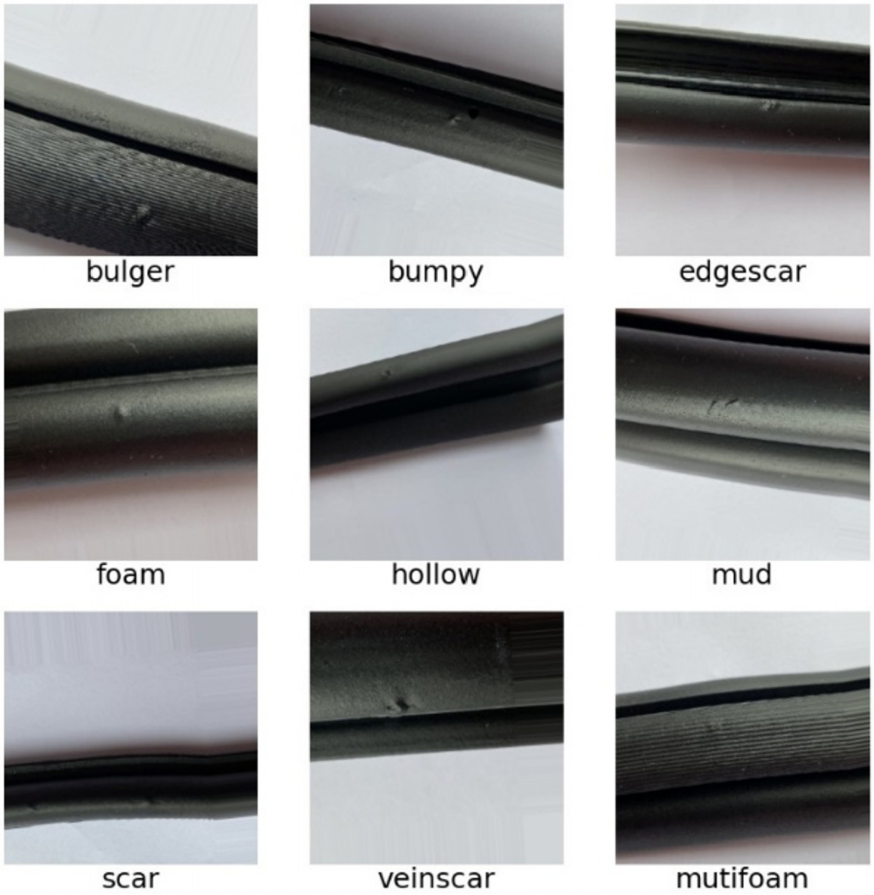

Figure 1. Different types of defects, products would produce such as lump and hollow. The sample size is small. There are a number flaw types on the surface of products, which affect product quality.

layers. By doing so, deep convolutional neural networks (CNN) could be trained on very deep networks (up to 152 layers) without falling into the vanishing gradient problem. The researchers also utilized transfer learning by pre-training the initial layers of the network on the large CIFAR100 dataset, allowing for faster convergence on smaller datasets. A pre-trained language model called BERT [5] (Bidirectional Encoder Representations from Transformers) was introduced, which uses a novel training objective to pre-train deep transformer networks on large textual datasets. The pre-trained BERT model can then be fine-tuned on smaller datasets for various NLP [6] (Natural Language Processing) tasks such as question answering and sentiment analysis. Transfer learning has been applied to a wide range of applications in recent research. A reinforcement learning-based approach [7] for learning transferable exploration strategies across graph-structured environments was proposed. By pre-training on diverse graph-structured datasets, the approach demonstrated improved performance on a wide range of unseen graph environments.

The data in the target domain [8] may have a different distribution or feature space than the data in the source domain. For example, the target categories in images are quite different from the source images. The images of defective auto parts encountered in engineering applications are shown in Fig. 1. The CIFAR100 dataset is used in the ResNet model. This can lead to a decrease in model performance, as the knowledge transferred from the pre-trained model may not be relevant to the target domain. To solve this problem, we propose an LDFISB (Large-scale Dataset to Few-Shot Image with Similar Background) algorithm. According to the characteristics of the defective image of auto parts, the proposed algorithm first removes the unnecessary part of the image, then increases the robustness of the image, and constructs new input images through rotation, affine, and other transformations. Finally, the transfer model achieves precision by fine-tuning, as shown in Figure 2.

Paper Organization: in Section 1, we introduce the problems encountered in transfer learning. In Section 2, we present the dataset and the proposed method. We demonstrate the novelty and the details of the LDFISB algorithm. In Section 3, we present related work and theory of normalization, boundary filling, etc. In Section 4, we show the details of the experiments and fine-tuning of LDFISB. In Section 5, we compare different models and the pre-training dataset and discuss the trends of loss and challenges. Following conclusion, we show the limitations and future perspectives of our method.

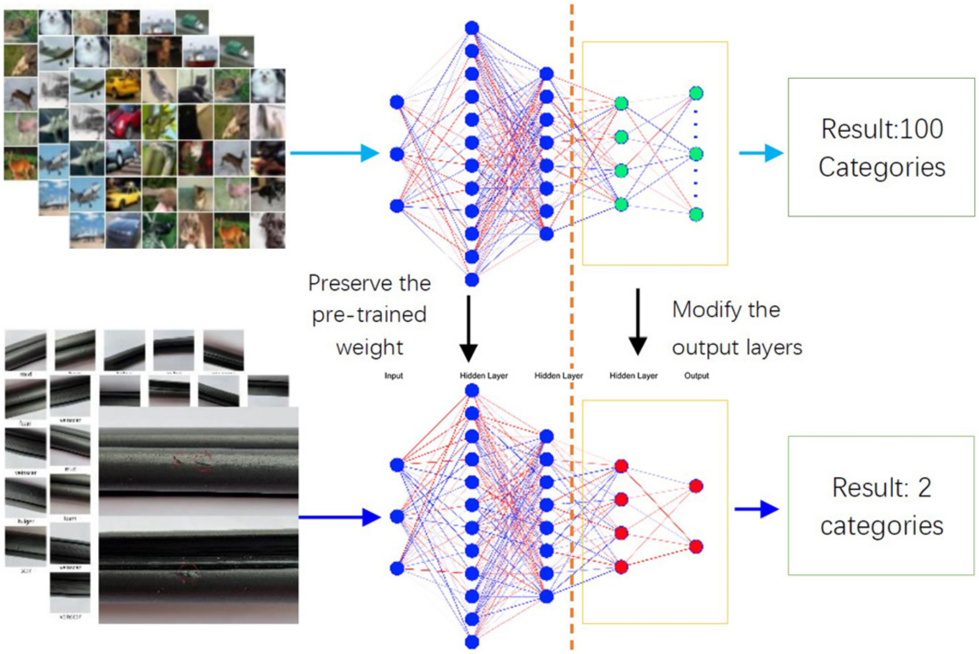

Figure 2. A framework for transfer learning of training parameters from CIFAR100 dataset to APSD (auto part surface dataset).

Table I. Introduction of APSD's categories.

Label	Introduction of categories
Defect/1	Multiple and single flaw objects exist on the product surface of input images, such as bumping, bulge, hollow, foam, and scar flaws.
Flawless/0	There are no defective area on the product surface of input images.

2. DATASET AND METHODS

2.1 *Dataset*

Due to the lack of the existing dataset to train in special surface detection, we processed and transformed the captured images to construct a flaw and flawless binary classification dataset, called APSD (auto part surface dataset). To expand the number of input images, a handful of the original images are transformed to generate a large number of generated images. Meanwhile, normalization [9] and boundary-filling [10] enhance the robustness of the model. APSD includes two categories of flaw and flawless products, as shown in Table I and consists of 5000 flawed images and 5000 flawless images. The ratio of the training dataset, validation dataset, and test dataset is 7:2.5:0.5.

2.2 *Method*

Transfer learning has become an important research area in the field of deep learning. With the ability to leverage pre-trained models on large datasets, transfer learning has the potential to improve generalization, reduce training times, and achieve state-of-the-art performance on a wide range of applications. Our approach (1) lends insights into features of these flaws (reasonable process and transformation), (2) outperforms previous methods without transfer learning, and (3) is robust to resizing, flipping, affine, rotation, and brightness. Transfer learning performs the neural network model from the large-scale dataset to few-shot [11] original images with similar background. The fine-tuned model achieves identifying flaw areas to predict the image classification. Our LDFISB algorithm is shown below.

The LDFISB Algorithm :

```
Input: The few-shot original images
Output: The result of the binary classification image
1: Expanding the number of input images: The few-shot original images are transformed to generate a large number of generated images.
2: Splitting into the dataset: The train dataset, validation dataset, and test dataset are 7:2.5:0.5.
3: Generating tensor(batch=64, 299, 299, 3)
4: Input a batch tensor image I: I →Transform (I) # Transform images
5: Load model and parameters # Neural network model: ResNet-50x1,101x1,101x3
6: Model(I)   # Fine-tuning the model as the scheduler and Weighted Cross Entropy Loss.
   6.1 Train dataset → Model(I)   # Apply a batch-splitting technique ("micro-batching")
   6.2 Validation dataset → Model(I)
   6.3 Test dataset t→ Model(I)
7: Saving the optimal parameters.
8: Predict (a new original image) → output (score) # The optimal model predicts a new original image to get the result of the binary classification.
```

3. RELATED WORK

3.1 *Normalization*

Normalization [12] of data is the scaling of data so that processed data falls into a small, specific range by Eq. (1). The weight of the neural unit will not change too much, which will not affect the performance of the model

$$\|X\|_p = ((|x_1|)^p + (|x_2|)^p + \cdots + (|x_3|)^p)^{\frac{1}{p}}. \tag{1}$$

Normalization can speed up convergence on models based on gradient descent or stochastic gradient descent. If the range of each feature dimension is different, the isocontour of the objective function is likely to be a group of ellipses. The greater the difference of each feature, the longer the ellipse contour will be. Since the direction of the gradient is perpendicular to the direction of the contour line. The route of the optimization will be more tortuous, so the iteration will be slow. In contrast, if the range of each feature dimension is similar, the objective function is likely to be close to a group of positive circles. The route of the optimization will be more direct and the iteration will be fast. In addition, normalization keeps the change of image eigenvalue unchanged, as shown in Figure 3.

It is indicated that the values on the y-axis have changed from [0–800] to [0–1]. The x-axis scale remains unchanged, and the feature curve of the image remains unaffected. Except for the value of the gradient changes, the image features do not change.

Batch Normalization is the normalization of the process on the batch, which is not effective for the small size of the batch. The data x_i is calculated by means and variances, and then normalized, as shown in Eq. (2).

$$\frac{x_i - \frac{1}{n}\sum_{i=1}^{n} x_i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 + \varepsilon}}. \tag{2}$$

LN (Layer Normalization) operates on the output of neurons in the specified layer. The mean and variance of the output of all neurons in this layer is calculated and then the output of this layer is normalized. LN is not effective on CNNs, but it is effective on RNNs.

To accelerate the convergence of the optimization process, WN is a normalization of the weights of the model by Eq. (3).

$$\omega = \frac{\mathrm{g}}{\|\mathrm{v}\|}\mathbf{V}. \tag{3}$$

V is a k-dimensional vector, g is a scalar, and ‖v‖ is the Euclidean norm of **V**. The Euclidean norm of the weight vector ω is fixed to g by Eq. (3), so that the activation of the neuron is approximately independent of **V**.

3.2 *Padding*

Convolution is the core operation of a CNN. The convolution methods of "edge pixels" of the image include "boundary padding before convolution" or "boundary padding after convolution" [13]. The boundary padding methods include constant padding, zero padding, mirror padding, and repeated padding and are described below. Zero padding: The Torch package adopts the ZeroPad2d function, and fills the Tensor using zero. A Tensor of the four directions is also filled by the padding parameter, such as a vector (1, 2, 3, 4). Constant padding: The torch package uses the ConstantPad2d function, which specifies the constant value as the padding. Zero padding is a special case of constant padding. Mirror padding uses the ReflectionPad2d function in the torch package. Compared with constant padding, the mirror padding method is likely to obtain better convolution results. The filled edge of the image is doubled in size. Repeated padding uses the ReplicationPad2d function in the torch package, which repeats the edge pixel value of the image, or extends the new boundary pixel value with the edge pixel value. The boundary pixel value after filling is a copy of the original pixel. The choice of padding method is more important for a small image but has little influence for a larger image.

3.3 *Cross Entropy Loss*

Cross entropy [14] can measure the different degrees between two different probability distributions in the same random variable. In the classification model, it is expressed as the difference between the true probability distribution and the predicted probability distribution. The smaller the cross-entropy value, the better the prediction of the model. The cross-entropy loss in binary classification is expressed as L, as shown in Eq. (4).

$$L = [y \log \hat{y} + (1 - y)\log(1 - \hat{y})]. \tag{4}$$

The $\hat{y}$ is the prediction distribution. The y is the real label.

There are several variations of cross-entropy loss [15], such as Cross Entropy Loss with Squared Loss, Sample Pairing Cross Entropy Loss, Boundary Cross Entropy Loss, Structured Cross Entropy Loss, and Weighted Cross Entropy Loss [16], and others. Boundary Cross Entropy Loss is used to solve the problem of extremely unbalanced data distribution. By setting the boundary of each category, the positive and negative samples are separated to reduce the misclassification of negative samples. Sample Pairing Cross Entropy Loss is often used in recommendation systems. By pairing positive and negative samples, the model can learn the features of both positive and negative samples in the training process. Structured Cross Entropy Loss is used to deal with sequential data or multi-label classification problems, which can consider the correlation between labels.

Weighted Cross Entropy Loss is commonly employed to address imbalances in categories, where certain categories appear more frequently than others in the dataset. By assigning different weights to different categories, the model can pay more attention to those categories that occur less frequently in the training process. The Weighted Cross Entropy Loss is capable of addressing the problem of class imbalance more effectively.

The defect areas of the samples are extremely different, even though they are relatively small. If Cross Entropy Loss is

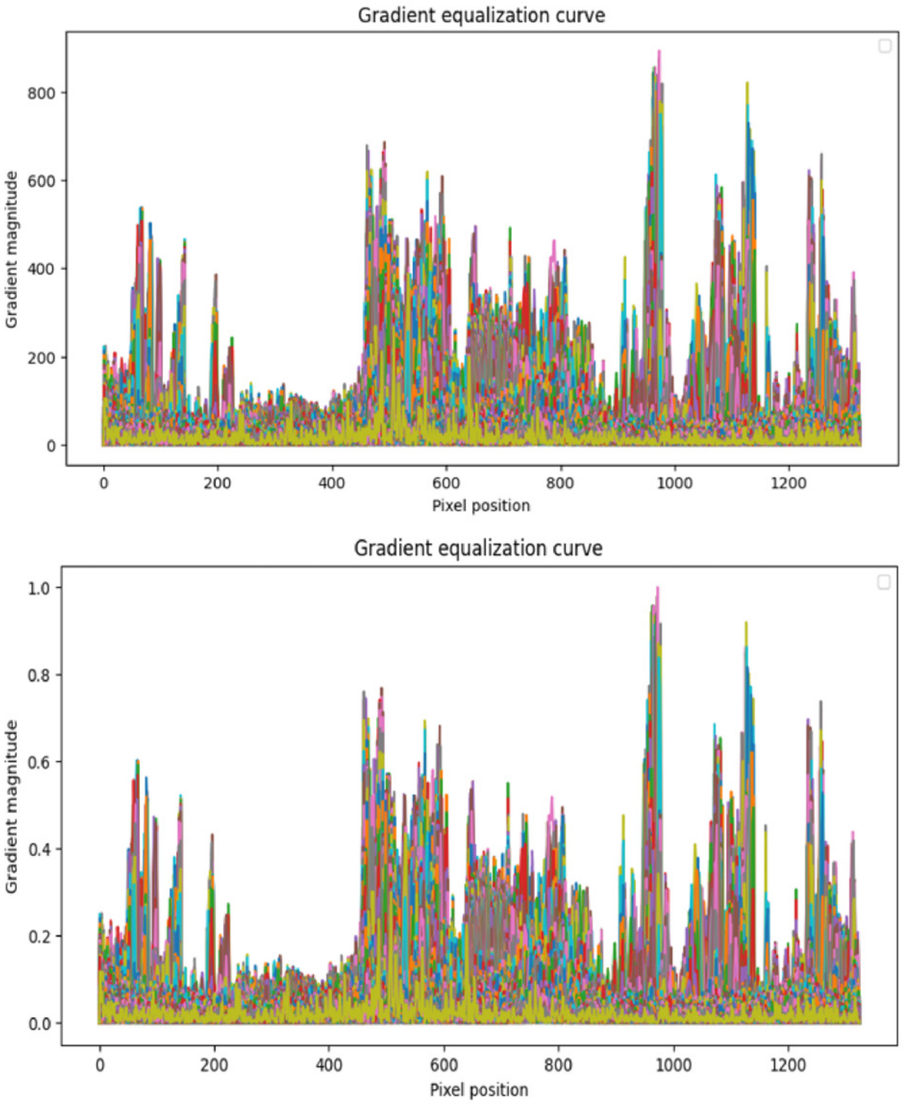

Figure 3. The changes in the feature gradient after normalization.

used, it is difficult to focus on the characteristics of the defect samples in the training. However, Weighted Cross Entropy Loss can solve this problem well and make the model pay more attention to the characteristics of defect samples. The Weighted Cross Entropy Loss function is shown in Eq. (5).

$$\mathrm{Lw} = [y * w_1 * \log \hat{y} + (1 - y) * w_2 * \log(1 - \hat{y})]. \qquad (5)$$

4. EXPERIMENTS

4.1 *Hyper-parameters*

Batch normalization, convolution kernel, sliding window, pooling, padding, Weighted Cross Entropy Loss, and other techniques are employed by the LDFISB in the training processes. Hyper-parameters are configured according to Table II to enhance the accuracy of the prediction by LDFISB. The initial LR (learning rate) is 0.003 and the momentum is 0.9. It determines how much the model's weights will change at each iteration and how the model can "remember" the direction of the gradient and continue moving in that direction even if the gradient changes slightly. A handful of the original images are transformed to generate a large number of generated images. Transformations of images include clipping, rotation, affine, brighten, and others. Three input images have been transformed from the original images, as shown in Figure 4. The APSD includes two categories of flaw and flawless products, which consist of 5000 flaw images and 5000 flawless images. The ratio of the training dataset, validation dataset, and test dataset is 7:2.5:0.5. The LDFISB model is trained for 800 steps, with a focus on observing the changes in the learning rate's decay at 100, 200, 300, 400, 500, and 600 epochs.

For the classification task of the flaw and flawless products, the LDFISB uses the SGD optimizer with an initial learning rate of 0.003, a momentum of 0.9, and an initial batch size of 16. The input image's tensor has a shape of (299, 299, 3). We adopt a simple heuristic fine-tuning

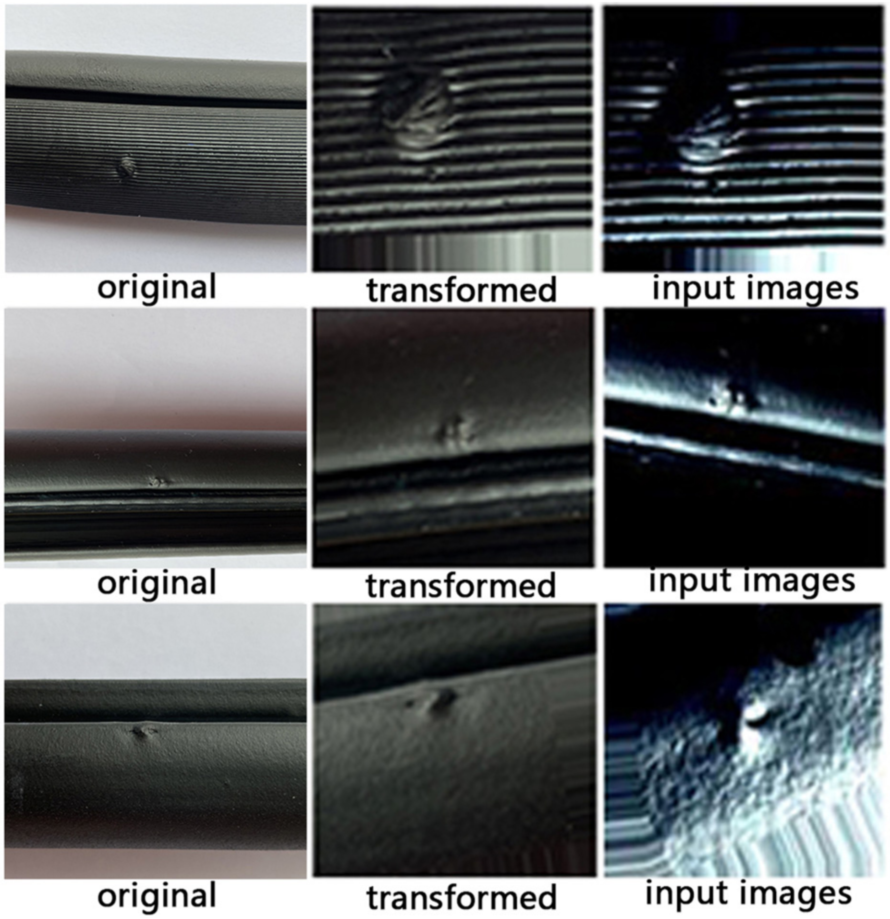

Figure 4. The original images generate the input images through transformation by clipping, rotation, affine, and brightness.

Table II. Hyper-parameters of the LDFISB.

Hyper-parameter	LR	Momentum	Batch	HSB	Total steps
Value	[3e-3, 3e-4, 3e-5, 3e-6, 3e-7]	[0.9, 0.09, 0.009, 0.0009]	[16, 32, 64, 128]	[100, 200, 300, 400, 500, 600]	[400, 500, 800, 1600]

strategy [17]. The configuration of HSB (hyper-parameter schedule boundaries) is [100, 200, 300, 400, 500, 600]. The LDFISB model can achieve perfect loss curves on the APSD dataset by achieving perfect loss curves through fine-tuning milestones.

4.2 *Training*

It is difficult to achieve perfect accuracy during training a few-shot object dataset by a CNN. CNN models are suitable for training a large-scale dataset to learn the features of images, but it does not generalize to few new examples of data. For example, when a handful of original images are directly trained on the ResNet-50 × 1 network, overfitting occurs, as shown in Figure 5.

Although the loss in training and validation sets show a downward trend in the whole epoch, it fluctuates too much at the beginning of the epoch. It shows that the data has a relatively large difference, and the change of features is relatively large when extracting the features of the image. The accuracy of training remained at [0.99, 1], and the accuracy of the test changed greatly at the beginning. Overfitting happens when the model starts using irrelevant features for prediction.

Our primary concern should be overfitting since we only have a few-shot object original image [18]. As the background of the images is very similar, it is difficult to obtain the features of the defect areas. The characteristics of the image affect the accuracy of the CNN model. The transfer learning for an image classification task is an appropriate attempt to train on our dataset. Our proposed LDFISB algorithm is a transfer learning method that uses the training parameters of the ResNet-50 × 1 network on the CIFAR100 dataset, and then fine-tuning the network to achieve the classification task. The result of training the APSD dataset is shown in Figure 6.

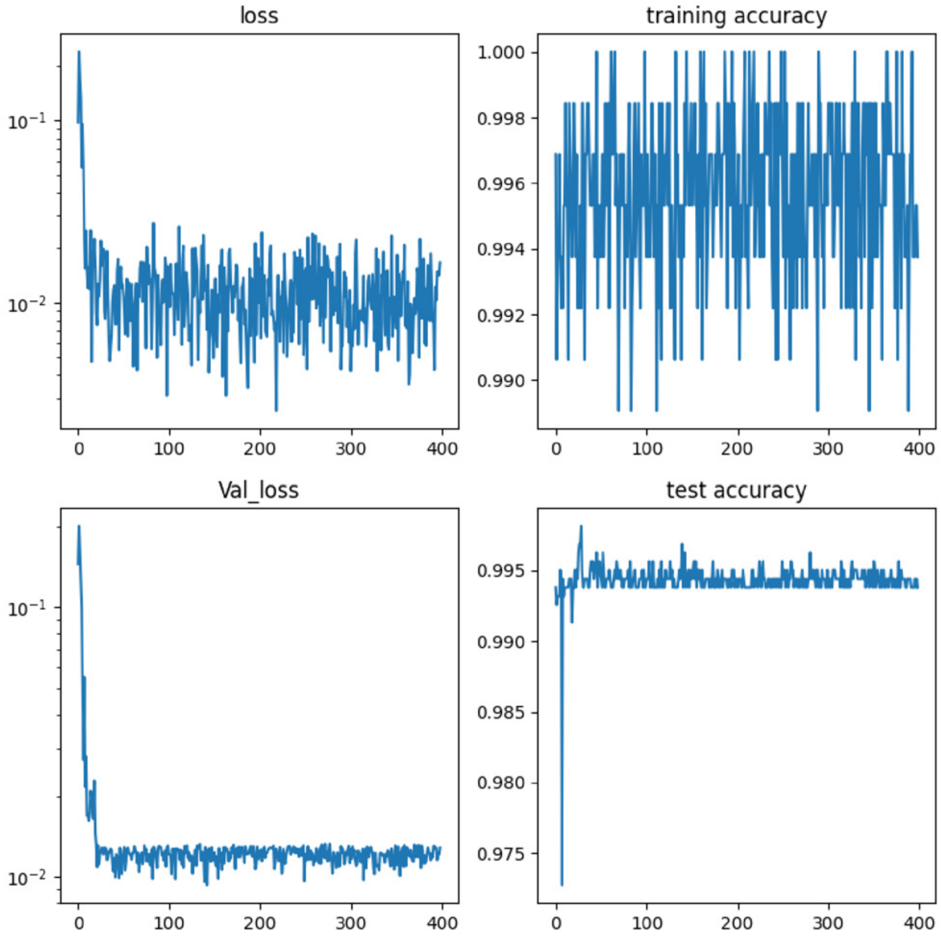

Figure 5. The ResNet-50 × 1 network directly trained the original dataset. The result shows that the change of loss is large, and the change of the accuracy is inconsistent.

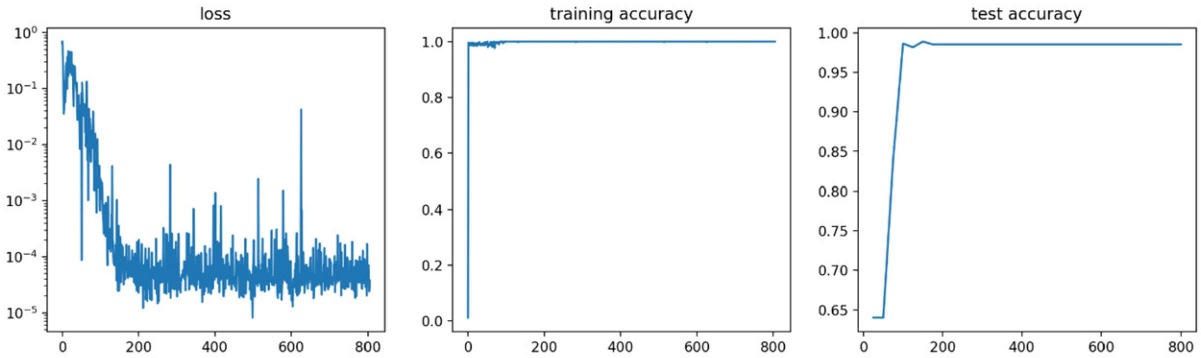

Figure 6. Training the APSD dataset on the ResNet-50 × 1 network by transfer learning.

Fig. 6 suggests that the training process is not ideal, although the accuracy of training and test sets are consistent. Data augmentation is an approach to combat overfitting, but it is not suitable because our augmented samples are still highly correlated.

The focus for overcoming overfitting should be the entropic capacity of the model, which is how much information is stored in the model. Using more features can improve the accuracy of a model that can store a lot of information, but storing irrelevant features is also a drawback. If the model is limited to storing only a few features, it must prioritize the most significant data that is more likely correlated, and have a better generalization.

The LDFISB method adopts different ways to modulate entropic capacity, i.e. batch normalization, regularization, boundary padding, and dropping. Regularization, such as L1 or L2 regularization [19], consists of forcing model weights to take smaller values. The LDFISB model uses few layers and

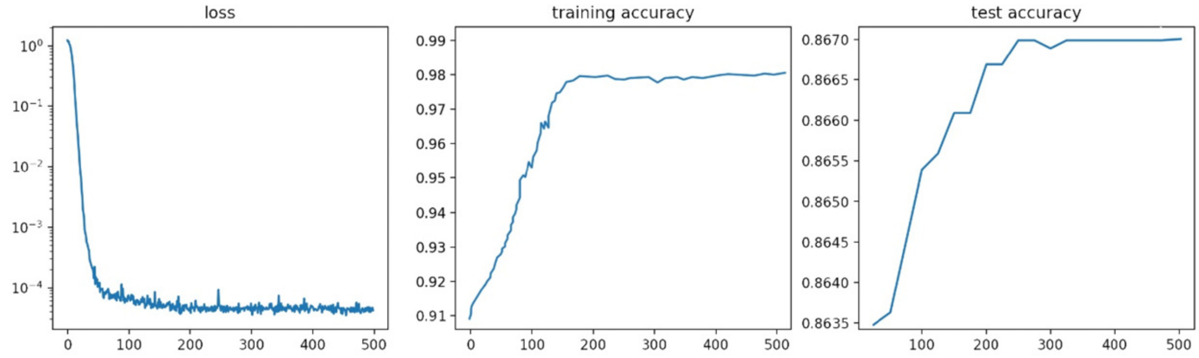

Figure 7. Training the APSD dataset on the ResNet-50 × 1 network by the proposed LDFISB algorithm.

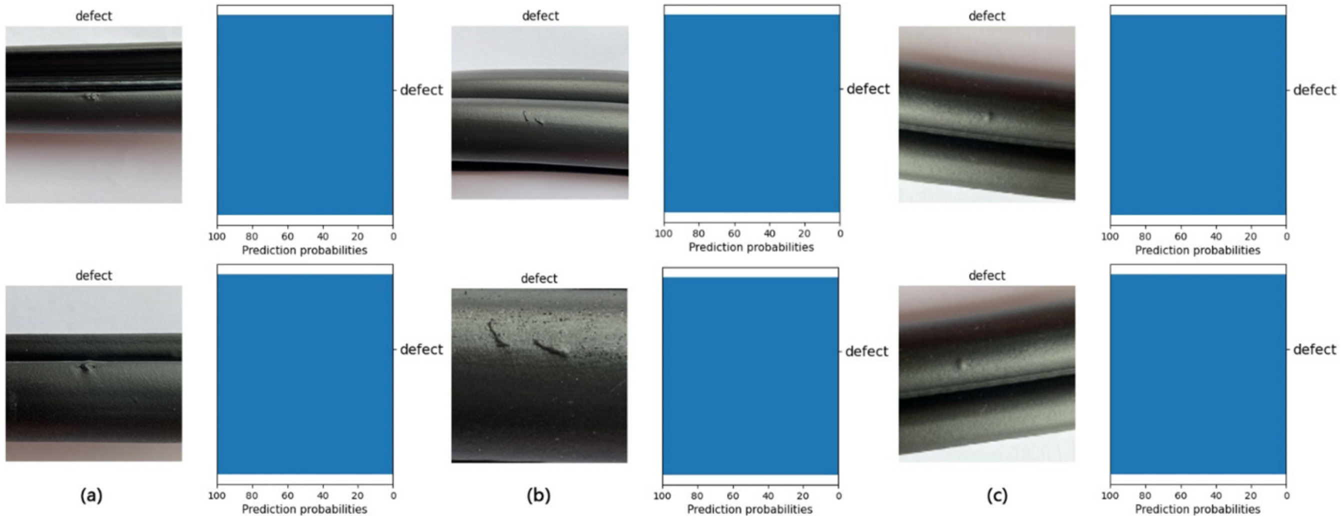

Figure 8. The LDFISB model predicts the different types of flawed product samples. (Group a with different types of flaws and 1860 × 729 sizes, Group b with the patch of the product surface, and Group c with several transformations).

few filters in the head layer, and data augmentation for input images. Dropout also helps reduce overfitting, by deleting the parameters to disrupt random correlations occurring in the training. According to the table of hyper-parameters, the optimal LDFISB model is obtained after several trainings. The loss and accuracy of the model are shown in Figure 7.

4.3 *Prediction*

We selected 3 groups of images, i.e. different types of flaws, a part image of the product surface, and a transformed image, for prediction by the LDFISB model and the results are shown in Figure 8. The original images of Group a with different types of flaws and 1860 × 729 sizes, Group b with the patch of the product surface, and Group c with several transformations tested the accuracy of the LDFISB model. The different original images can fully verify the robustness of the model. The results of the test show that the LDFISB model is perfect for distinguishing between flawed and flawless products.

5. COMPARISON AND DISCUSSION

5.1 *Comparison*

The LDFISB is tested to generalize across three pre-training datasets, i.e. CIFAR100, CIFAR10, and ImageNet. The transfer model employs three different Backbone networks, i.e. ResNet-50 × 1, ResNet-101 × 1, and ResNet-101 × 3. According to the APSD dataset, the model is devised with more efficient application-specific hyper-parameters. For instance, we tested the LDFISB model using an RTX GPU machine on the CIFAR10, CIFAR100, and ImageNet datasets [20], while increasing batch size from 16 to 128 and LR from 0.003 to 0.001. Three ResNet models are used to compare the different datasets. The default Hyper-parameters were developed in the training process to achieve the optimal model. To exploit the desired accuracy, it is necessary to adjust the LR and schedule (steps). Meanwhile, the code applies a batch-splitting technique (“micro-batching”) to reduce memory requirements. The results of comparison obtained by different models on different datasets are shown in Table III, which shows that

Table III. The LDFISB model is tested to generalize across three pre-training datasets by the transfer model.

Transfer model	Pre-training	Accuracy
ResNet-50 × 1	ImageNet	85.49%
ResNet-101 × 1	ImageNet	85.87%
ResNet-101 × 3	ImageNet	86.26%
ResNet-50 × 1	CIFAR10	84.52%
ResNet-101 × 1	CIFAR10	84.12%
ResNet-101 × 3	CIFAR10	83.87%
ResNet-50 × 1	CIFAR100	86.70%
ResNet-101 × 1	CIFAR100	85.65%
ResNet-101 × 3	CIFAR100	85.27%

the model of transfer learning on the CIFAR100 pre-training dataset achieves the highest accuracy.

5.2 *Discussion*

Transfer learning involves transferring knowledge learned from a pre-trained model on one task to a different but related task. While transfer learning has shown great promise in improving the accuracy and efficiency of neural network models, there are limitations and challenges. One major problem with transfer learning is domain shift. Domain shift occurs when there is a difference between the data distribution of the source domain, where the pre-trained model was trained, and the target domain, where the model is being applied. The data in the target domain have a different distribution or feature space than the data in the source domain. This can lead to an overfitting in model performance, as the knowledge transferred from the pre-trained model may not be relevant to the target domain. For example, if the original image is directly trained on the ResNet model, overfitting will occur. Another challenge in transfer learning is task transferability. While some tasks may be highly related and transferable, others may not. For example, a pre-trained model on common object image tasks may not be as transferable to uncommon object images with similar background. Therefore, finding the right pre-trained model for a given task can be a challenging task in itself.

Transfer learning can also suffer from negative transfer. Negative transfer [21] occurs when the pre-trained model has learned features or patterns that are not relevant to the target task and harms the model's performance. This is often due to differences in the data distribution or feature space between the source and target domains.

Furthermore, there is the issue of ethical concerns in transfer learning. Pre-trained models may have been trained on biased or unethical data, and transferring this knowledge to a new task may perpetuate biases or ethical concerns in the new domain. Therefore, it is important to carefully consider the ethical implications of transfer learning and apply it responsibly.

6. CONCLUSIONS

Deep learning usually requires large amount of data to train a model to complete the task. Learning the feature of few-shot object images is a challenge. Although original images are few, our proposed LDFISB method provides a feasible solution. This model is trained on a large-scale dataset such as CIFAR100, and then the model is fine-tuned based on the original model and parameters to achieve classification tasks on a new APSD dataset. The proposed model of transfer learning can enhance the accuracy of the prediction, which accurately predicts the flaw area of the product image. Moreover, it achieves the highest accuracy on the CIFAR100 pre-training dataset. While transfer learning has shown great potential in improving the performance of machine learning models, it is not without its limitations and challenges. Domain shift, task transferability, negative transfer, and ethical concerns are just a few of the issues that need to be carefully considered when applying transfer learning techniques.

ACKNOWLEDGMENT

This research was supported by National Nature Science Foundation of China (No. 62262006, No. 52375270), State Key Laboratory of Geo-Information Engineering and Key Laboratory of Surveying and Mapping Science and Geospatial Information Technology of MNR, CASM (No. 2023-04-03), Technology Innovation and Application Development Key Project of Chongqing (No. cstc2021jscx-gksbX0058), Science and Technology Project of Henan Province (No. 232102210103), The research project of Xuchang University (No. 2024YB015). Henan Science and Technology Research Project (No.1040).

REFERENCES

[1] X. Yue, G. Ma, F. Liu, and X. Gao, "Research on image classification method of strip steel surface defects based on improved Bat algorithm optimized BP neural network," J. Intell. Fuzzy Syst. **41**, 1509–1521 (2021).

[2] A. Brodzicki, M. Piekarski, D. Kucharski, J. Jaworek-Korjakowska, and M. Gorgon, "Transfer learning methods as a new approach in computer vision tasks with small datasets," Foundations Comput. Decision Sci. **45**, 179–193 (2020).

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 770–778.

[4] A. Fazari, O. J. Pellicer-Valero, J. Gómez-Sanchıs, B. Bernardi, S. Cubero, S. Benalia, G. Zimbalatti, and J. Blasco, "Application of deep convolutional neural networks for the detection of anthracnose in olives using VIS/NIR hyperspectral images," Comput. Electron. Agricult. **187**, 106252–106261 (2021).

[5] A. Pardamean and H. F. Pardede, "Tuned bidirectional encoder representations from transformers for fake news detection," Indonesian J. Electr. Eng. Comput. Sci. **22**, 1667–1671 (2021).

[6] S. Gombert, D. Di Mitri, O. Karademir, M. Kubsch, H. Kolbe, S. Tautz, A. Grimm, I. Bohm, K. Neumann, and H. Drachsler, "Coding energy knowledge in constructed responses with explainable NLP models," J. Comput. Assisted Learning **39**, 767–786 (2022).

[7] Y. Wang, W. Zhang, M. Hao, and Z. Wang, "Online power management for multi-cores: a reinforcement learning based approach," IEEE Trans. Parallel Distributed Syst. **33**, 751–764 (2022).

[8] X. Liang, Y. Zhang, and J. Zhang, "Attention multisource fusion-based deep few-shot learning for hyperspectral image classification," IEEE J. Select. Top. Appl. Earth Observations Remote Sens. **14**, 8773–8788 (2021).

[9] G. Li, Z. Chen, Z. Yang, and J. He, "Novel learning functions design based on the probability of improvement criterion and normalization techniques," Appl. Math. Modell. **108**, 376–391 (2022).

[10] S. Li, J. Wang, L. Li, S. Shi, and Z. Zhou, "The theoretical and numerical analysis of water inrush through filling structures," Math. Comput. Simulat. **162**, 115–134 (2019).

[11] J. Y. Lim, K. M. Lim, S. Y. Ooi, and C. P. Lee, "Efficient-PrototypicalNet with self knowledge distillation for few-shot learning," Neurocomputing **459**, 327–337 (2021).

[12] J. Sun, X. Cao, H. Liang, W. Huang, Z. Chen, and Z. Li, "New interpretations of normalization methods in deep learning," *Proc. AAAI Conf. on Artificial Intelligence* (AAAI, Washington, DC, 2020), Vol. 34, pp. 5875–5882.

[13] X. Zhao, J. Jiang, K. Feng, B. Wu, J. Luan, and M. Ji, "The method of classifying fog level of outdoor video images based on convolutional neural networks," J. Indian Soc. Remote Sens. **49**, 2261–2271 (2021).

[14] Y. Lv, B. Qian, R. Hu, H. P. Jin, and Z. Q. Zhang, "An enhanced cross-entropy algorithm for the green scheduling problem of steelmaking and continuous casting with uncertain processing time," Comput. Industr. Eng. **171**, 1–22 (2022).

[15] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," IEEE Trans. Vehicular Technol. **68**, 4204–4212 (2019).

[16] P. Zhong, D. Wang, and C. Miao, "An affect-rich neural conversational model with biased attention and weighted cross-entropy loss," *Proc. 33rd AAAI Conf. on Artificial Intelligence* (AAAI, Washington, DC, 2019).

[17] X. Wu, R. Lian, D. Jiang, Y. Song, W. Zhao, Q. Xu, and Q. Yang, "A phonetic-semantic pre-training model for robust speech recognition," CAAI Artificial Intell. Res. **1**, 1–7 (2022).

[18] M. Xiao, Y. Wu, G. Zuo, S. Fan, H. Yu, Z. A. Shaikh, Z. Wen, and D. M. Shafiq, "Addressing overfitting problem in deep learning-based solutions for next generation data-driven networks," Wireless Commun. Mobile Comput. **2021**, 1–10 (2021).

[19] S. C. Prasad and S. Balasundaram, "On lagrangian L2-norm pinball twin bounded support vector machine via unconstrained convex minimization," Inform. Sci. **571**, 279–302 (2021).

[20] K. Nakai, T. Matsubara, and K. Uehara, "Neural architecture search for convolutional neural networks with attention," IEICE Trans. Inform. Syst. **E104.D**, 312–321 (2021).

[21] T. Keil, D. Lavie, and S. Pavićević, "When do outside CEOs underperform? From a CEO-centric to a stakeholder-centric perspective of post-succession performance," Acad. Manage. J. **65**, 1424–1449 (2022).