Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 1903292, 14 pages
https://doi.org/10.1155/2023/1903292

Research Article

Automobile Component Recognition Based on Deep Learning Network with Coarse-Fine-Grained Feature Fusion

Jinbiao Tan, Jiafu Wan, and Dan Xia

School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China

Correspondence should be addressed to Jiafu Wan; mejwan@scut.edu.cn

Received 17 September 2022; Revised 20 January 2023; Accepted 30 January 2023; Published 21 February 2023

Academic Editor: Vittorio Memmolo

With the development of artificial intelligence, machine vision technology based on deep learning is an effective way to improve production efficiency. Because of the rapid update of the automobile manufacturing industry and the large variety of products, the learning time and the number of learning samples of the deep learning model are limited, which brings great difficulties to the recognition of components. Therefore, considering the economic benefits of enterprises, this paper proposes an intelligent component recognition method appropriate for small datasets, aiming to explore an automatic system for component recognition suitable for industrial manufacturing environments. The method completes the generation of the dataset through the system architecture with the potential for automation and the image cropping method based on feature detection and then designs a deep learning network based on coarse-fine-grained feature fusion to generate an intelligent recognition model of components. Finally, the designed network achieves an accuracy of 95.11%, and compared with the traditional classical network on multiple datasets, the designed network has better performance. Thus, the proposed method can improve the production flexibility of the automobile manufacturing industry and improve equipment intelligence.

1. Introduction

With the widespread application of artificial intelligence technology in industrial manufacturing, the demand for automation in the automotive assembly manufacturing industry is increasing. Target detection and recognition based on deep learning are an important technical means to promote equipment intelligence and production automation [1]. However, the current rapid update and many types of products in the automotive manufacturing industry bring challenges to the intelligent recognition of components. The fast update speed requires that the production of datasets for deep learning consume low time costs, while the many product types command the intelligent recognition model to have strong robustness and accuracy and be able to recognize different types of components. Therefore, in response to the above problems, figuring out how to design a rapid production method for component datasets and building a deep learning network suitable for small datasets are the key to improving the automation of automobile manufacturing.

Today, the rise of new energy vehicles has enriched the types of vehicles, and the styles of components have increased geometrically. The traditional method of pasting barcodes on components is inefficient and lacks flexibility, making it difficult to adapt to the needs of intelligent manufacturing in the new era. Deep learning technology has excellent performance in target detection [2] and image recognition [3] and can be used to give equipment the ability to automatically recognize targets and enhance the intelligence and flexibility of the equipment. Nowadays, there are many deep learning models (e.g., VGG [4], ResNet [3], Fast-RCNN [5], and YOLO [6]) that are widely used for production defect detection [7], product quality control [8], and object recognition [9–11]. But these models have similar characteristics, that is, they require many learning samples to gain experience. For Faster-RCNN and YOLO, it also takes a long time to label images with "LabelImg." In the fast-updated automobile manufacturing industry, it is difficult to obtain enough learning samples to support the learning of deep learning models, and cumbersome data annotation will

also increase the production cost of manufacturing companies, making the applicability of these deep learning models limited.

Thus, aiming at the particularity of the automobile manufacturing industry, this paper desires to explore an intelligent recognition method for components suitable for industrial manufacturing. Focusing on the production cost and efficiency of enterprises, this paper explores a reliable and lightweight intelligent system to realize the automatic generation of component datasets, as well as the automatic training, deployment, and upgrading of models. The main contributions of this paper are as follows:

(1) An intelligent recognition architecture for auto components with automation potential is proposed, which includes three layers of data acquisition, deep learning, and model application. The automatic implementation of "data collection-network learning-model deployment-model upgrade" can be completed.

(2) In the context of small sample data, an intelligent recognition method of auto components based on a parallel deep learning network (PDLN) is proposed. This method obtains a reliable recognition model when the learning samples are insufficient by fusing coarse- and fine-grained features.

(3) Combined with the image feature detection algorithm, an image target cropping method is designed, which can be used to speed up the generation of datasets, and improve the robustness of the model application.

The rest of this paper is organized as follows: Section 2 provides a review of related work. Section 3 presents the overall system architecture. Section 4 describes the feature detection-based image cutting method. The PDLN's design is shown in Section 5. Section 6 reports the experimental process and discussion. Finally, Section 7 provides discussion. Section 8 concludes the paper.

2. Related Work

Using deep learning technology to assist industrial production is an effective way to improve production intelligence. In particular, the use of image recognition algorithms to give equipment the ability to recognize production content can better realize intelligent manufacturing. Different from traditional image recognition algorithms, intelligent image recognition based on deep learning has better robustness and is currently widely used in product quality monitoring [8] and object recognition [3]. As shown in Table 1, image recognition algorithms can be roughly divided into two categories, one is that an image has only one recognized target, and the other is that an image contains multiple recognized targets. The difference between the two is that the former model is relatively simple and data preprocessing (data labeling) is efficient, while the latter model is usually more complex and data preprocessing is relatively cumbersome.

Considering the rapid update speed of automotive products and the large number of components, using a recognition method that only contains a single recognized target in an image can avoid the increase in cost caused by the manual data annotation described in [12, 13, 16] and is more in line with the efficiency requirements of manufacturing enterprises. For the problem of insufficient learning samples, some researchers propose to use Gan [18] to increase sample data. However, this will increase costs and time consumption. Therefore, it is necessary to design a deep learning model suitable for a small number of learning samples, and there are still some shortcomings in the current research methods. For example, a method that can realize the whole process automation of "dataset production-network training-model deployment-model upgrade" can maximize the improvement of production efficiency and optimize the manufacturing mode. Thus, the method designed in this paper will maximize the automation of the whole process.

3. System Architecture

Automation and intelligence are important symbols of the new generation of industrial manufacturing and important guarantees for reducing labor and improving manufacturing efficiency. This paper proposes an intelligent recognition architecture for auto components with automation potential, as shown in Figure 1. The architecture consists of three layers: data collection, deep learning, and model application. From left to right, each layer is the basis for the next layer.

3.1. Data Collection. Abundant data is an important foundation for deep learning network training. In industrial manufacturing such as automobile assembly, the image data acquisition of components can automatically obtain rich image samples by installing camera equipment on the finished components. Secondly, it is also possible to manually take images of relevant components by workers. Both abovementioned methods have their limitations. For example, the images of components automatically captured at a fixed position are relatively simple, and the styles are not rich enough (the first row of Figure 2). Although manual shooting can make up for the abovementioned shortcomings, it needs to consume a lot of labor, which is not conducive to the development of enterprises. Therefore, in automobile assembly manufacturing, it is often more economical to build small datasets.

Furthermore, with a background in personalized customization, the production mode of small batches promotes faster product iteration and more complex product styles. This limits the feasibility of making large datasets and requires that datasets be manufactured faster. Thus, based on the image feature detection algorithm, this paper designs an object cropping method, which can speed up the production of datasets and realize automatic generation. For details, please refer to the fourth section.

Table 1: Deep learning-based image recognition in industry.

Ref.	Concept	Methodology	One image		Preprocessing	Small dataset	Performance
			Single target classification	Multiobject segmentation			
[7]	Surface defect detection	CNN + Unet		√	—	No	—
[12]	Weld defect classification	VGG16 + SegNet-modified	√		Manually annotated	—	—
[13]	Subsurface defects detection	YOLO v5		√	Manually annotated	—	Precise = 92.6%
[14]	Quality monitor of pizza packages	ResNet18	√		Image flip	No	Precise = 99.74%
[15]	Recognition of mechanical material structure	Deep learning network	√		—	Yes	Accuracy = 90.8%
[16]	Vehicle types detection	G-YOLOX		√	Manually annotated	—	Precise = 95.0%
[17]	Damage detection of grotto murals	Ghost-C3SE YOLOv5		√	Manually annotated	Yes	Precise = 56.74%
This paper	Component recognition	PDLN	√		None	Yes	95.11%

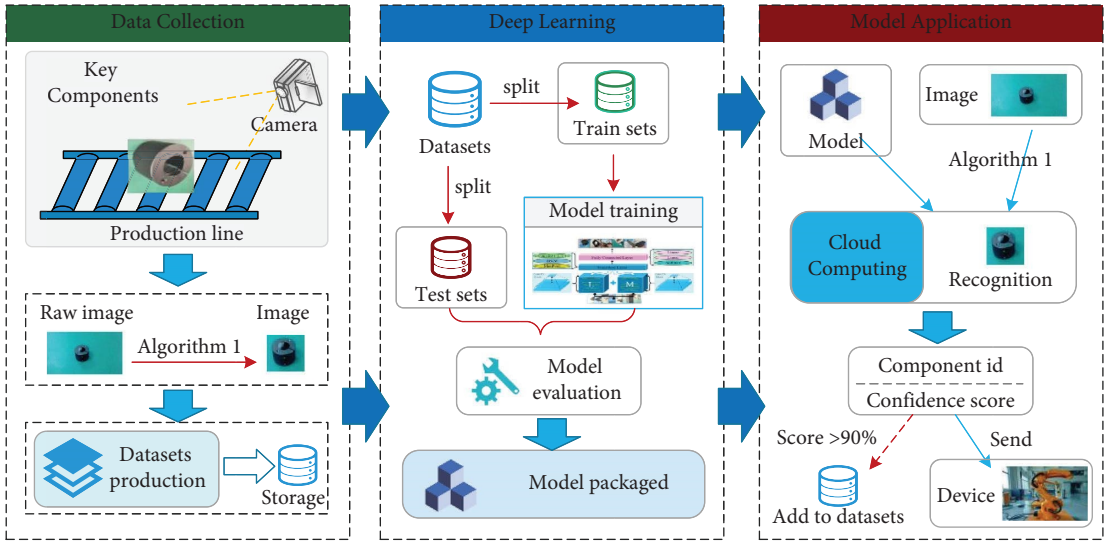

Figure 1: Intelligent recognition architecture for auto components.

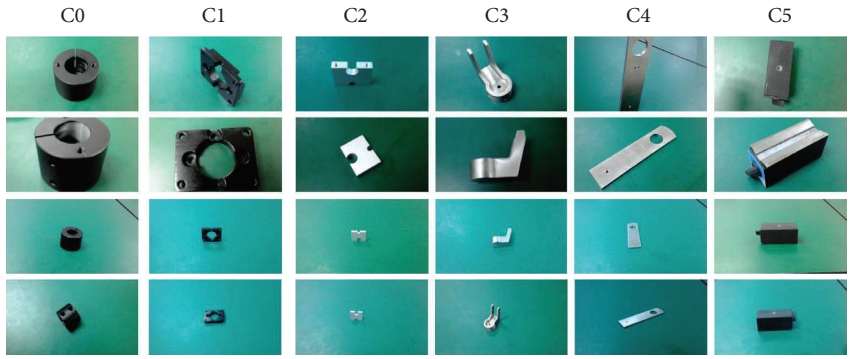

Figure 2: Samples of DATA_DEVICE.

3.2. Deep Learning. Deep learning technology has outstanding performance in image classification and recognition and has higher accuracy than traditional image algorithms. In image classification and recognition, the classic deep learning networks include VGG, ResNet, and DenseNet [19], which usually require a large number of learning samples to obtain reliable model performance. However, the small datasets in the automobile assembly manufacturing industry make it difficult to support the training of the abovementioned deep learning networks, and it is impossible to obtain reliable model performance. Therefore, designing a deep learning network for small datasets is the key to establishing intelligent recognition of auto components. In this paper, based on coarse-fine-grained analysis, a parallel deep learning network is designed for intelligent component recognition of small datasets.

3.3. Model Application. The trained deep learning network can generate an intelligent recognition model through the storage mechanism of *Pytorch*. Because of the limitations of the computing power of the production equipment itself, the model can be deployed in the cloud. Thus, the images of the component, which are transmitted to the cloud, can be recognized by the model in the cloud. Then, the recognition result will return to the equipment, as shown in Figure 3. In addition, the viewing angle of the equipment for shooting components may be too large, resulting in a small proportion of the image of the components (as shown in the third row in Figure 2) and a decrease in the accuracy of model recognition. Therefore, it is necessary to design an algorithm for intelligent cropping. Like crafting datasets, feature detection-based object cropping methods can play an important role in this process. In addition, during the application process, images with recognition confidence greater than 90% will be added to the dataset for reinforcement learning in subsequent models. In this way, the performance of the model is continuously improved, and the automation of data acquisition, model training, and model upgradation is completed.

4. Feature Detection-Based Object Cropping

In the original image of the camera, the object area occupies relatively little of the total image area, resulting in a waste of computing resources and a lack of focus. So, cropping should be done to the original image to highlight the object and reduce the image size. However, cropping out the object

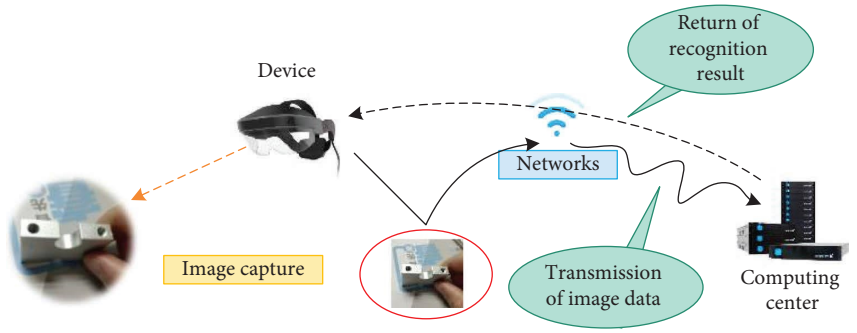

Figure 3: Ways of model application.

from each image often takes a lot of time and effort, which may reduce production efficiency. Thus, a novel method based on feature detection that can automatically crop images is the key to speeding up the generation of component datasets and is more suitable for fast iteration production methods.

4.1. Key Methods

4.1.1. Bilateral Filter. Bilateral filtering [20] is a nonlinear filtering method that combines spatial proximity and image pixel similarity values. More precisely, this method considers spatial information and grayscale similarity. It has a good preservation effect on the image's contour edge and can eliminate the speckle noise inside the contour simultaneously. Its core formula is as follows:

$$\begin{cases} g(i,j) = \dfrac{\sum_{(k,l)\in S(i,j)} f(k,l).w(i,j,k,l)}{\sum_{(k,l)\in S(i,j)} w(i,j,k,l)}, \\ w(i,j,k,l) = \exp\left(-\dfrac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \dfrac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right), \end{cases} \tag{1}$$

where $g_{(i,j)}$ denotes pixel$_{(i,j)}$'s value and $s_{(k,\ l)}$ refers to pixels within a $(2n+1)$ range from pixel$_{(i,j)}$ (i.e., pixel$_{(i,j)}$ acts as a center, $f_{(k,l)}$ contains the position pixel$_{(k,\ l)}$, s weight, and $w_{(i,j,k,l)}$ is the value calculated using two Gaussian functions. The weights related to the pixel distance and similarity are denoted σ_d^2 and σ_r^2, respectively. The σ_r^2 parameter preserves the image boundary information.

4.1.2. Gaussian Filter. Bilateral filtering could leave the impulse noise [21, 22]. Therefore, further noise reduction and image smoothing are necessary. A Gaussian filter is an algorithm that convolves the image utilizing a Gaussian kernel, i.e., a coordinate system (x, y) that follows certain rules. The center pixel's value is obtained by weighting and summing the neighboring pixels' values, as given in

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\pi\sigma^2}\right). \tag{2}$$

4.1.3. ORB Feature Detection. Oriented FAST and rotated BRIEF (ORB) feature detection [23] is performed on single-channel grayscale images. ORB judges whether a corner pixel and pixel$_{(x,y)}$ are feature points by detecting the number of pixels in the pixel$_{(x,y)}$'s neighborhood whose value differs from pixel$_{(x,y)}$'s by more than h. The pyramid algorithm enables the feature detection's scale invariance. In addition, the ORB algorithm assumes a certain offset between the corner pixel's grayscale and the centroid, and a characteristic orientation can be obtained by calculation. We define the moment of the corner pixel$_{(p,q)}$'s neighboring pixels as

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y), \tag{3}$$

where $I_{(x,y)}$ denotes pixel$_{(x,y)}$'s gray value. The image centroid is now obtained as

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right). \tag{4}$$

The angle between the feature point's position and the centroid is defined as the feature point's orientation:

$$\theta = \arctan\left(m_{01}, m_{10}\right). \tag{5}$$

To improve the method's rotation invariance, it is necessary to ensure that x and y are contained in a circular area with radius r (i.e., $x, y \in [-r, r]$), where r is the neighborhood radius. These methods enable obtaining a large amount of feature point information in the image.

4.2. Object Areas' Intelligent Cropping. As different camera sensors have different sensitivity to light and color, the image information of objects in the same scene captured by various camera sensors is not consistent. There are many Gaussian and "pretzel" noises in the image, which can affect the feature detection, misleading the location of the object. Therefore, we use filtering algorithms (e.g., bilateral filtering algorithm [20] and Gaussian filtering algorithm [21, 22]) to smooth the image for removing the noise in the image. With the noise reduced significantly, the number of feature points detected by ORB will be sacrificed, while these points will be more concentrated on the object. As shown in Figure 4, after

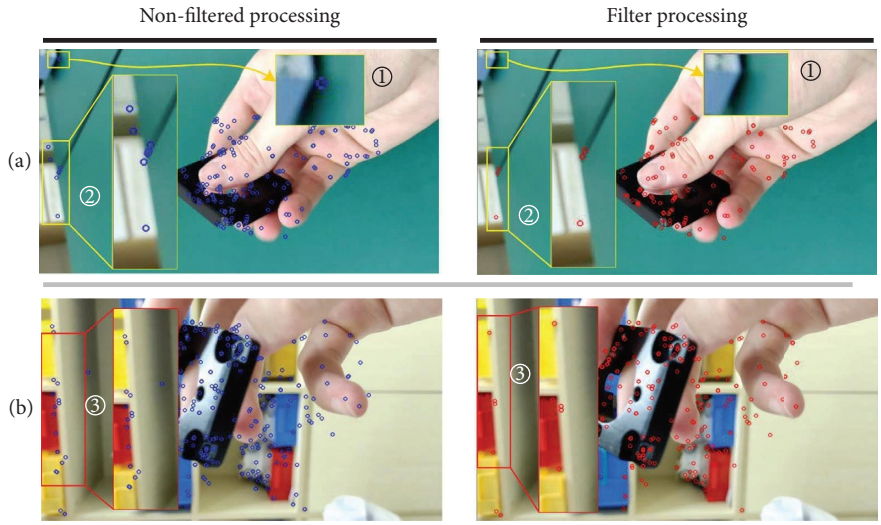

Figure 4: The influence of filter processing in feature detection.

the image is smoothed by noise reduction, the feature points are mainly distributed near the object area, which helps to obtain a smaller size image of the object.

This study utilizes the ORB feature detection algorithm to obtain the positions of all feature points in the image. The position coordinates are denoted as

$$T = \{(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots (x_{n-1}, y_{n-1})\}. \quad (6)$$

According to multiple tests, the area of an object (component) in the image can be located with these feature points' help. Then, a square box is used to surround these feature points, and then the square feature area is cropped to generate a smaller size image of the components. The generation process is shown in Algorithm 1.

The cropping area is k times the feature area. This study has tried many times to choose a proper value of k so that the cropping area can completely cover the object. In the end, a good value of k is chosen: $k = 1.2$. As shown in Figure 5, the feature detection algorithm can effectively crop into a smaller size and more concentrated information image of the object, from the original image, which helps to reduce the computational cost of the deep learning network.

Finally, a dataset containing six types of components is obtained. For each type, images with multiple backgrounds and scenes are collected. As shown in Figure 6, the components' background is complex and diverse, which reflects the actual production environment's complexity. This comprehensive and realistic dataset consists of 2,040 images in total.

5. Parallel Deep Learning Network

How to obtain reliable deep learning models on small datasets is still a difficult problem, today. This study proposes an image coarse-fine-grained feature fusion method to improve the learning ability of deep learning networks for image features and help networks get more image features.

5.1. Coarse-Fine-Grained Feature Fusion Architecture. ResNet and DenseNet effectively solve the gradient descent problem in deep networks by using a residual structure, which results in a huge improvement in network depth. The multilayer convolutional block of VGG has a strong feature extraction capability. Studies show that under the same convolution kernel, the receiving field of the deep convolutional network is larger but with less information, while the receiving field of the shallow convolutional network is smaller but with more information.

Thus, we propose an image coarse-fine-grained feature fusion method to make deep learning networks that can obtain the coarse-grained and fine-grained features of images. Thus, networks will have enough features to learn without big datasets and receive more detailed and global information. The method consists of a novel network architecture, as shown in Figure 7, named parallel deep learning network (PDLN).

By stacking different numbers of convolutional layers, the PDLN designs two convolutional links (large and minor, L and M) of a large receptive field and a minor receptive field and obtains the global features and local features of the input image, which are fused and input to the fully connected layer to achieve classification and recognition. Various features of the input tensor will be extracted from convolutional links with large and minor receptive fields and achieve fine-grained recognition at a low depth of the network. The PDLN does not obtain the large receptive field by deepening the network, which effectively saves the local feature of minor receptive fields for the image and avoids the overfitting problem.

To better understand the principle of our architecture, we visualize the learning process of PDLN, and we can find that the input image features extracted by the two links (L and M) of PDLN are significantly different. As shown in Figure 8, both L and M map an input image to more dimensions to extract features (to get different feature vectors in multiple dimensionalities). The feature maps extracted by

```
(01) begin
(02)     input ← input an image
(03)     imageFiltering() //filtering noise reduction
(04)     Detect all key points
(05)     for (iterate through all key points)
(06)         Remove relatively isolated points
(07)         Statistics key points scope
(08)     end for
(09)     Use a square to enclose the scope
(10)     if (the square is beyond the bounds of the image = false)
(11)         Increase the square by a factor of k
(12)     end if
(13)     Cut out the square to form a smaller picture
(14)     Save this picture
(15) end
```

Algorithm 1: Image cropping.

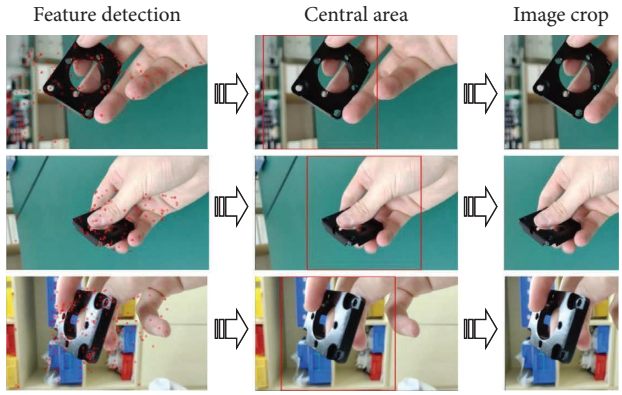

Figure 5: Dataset production scheme.

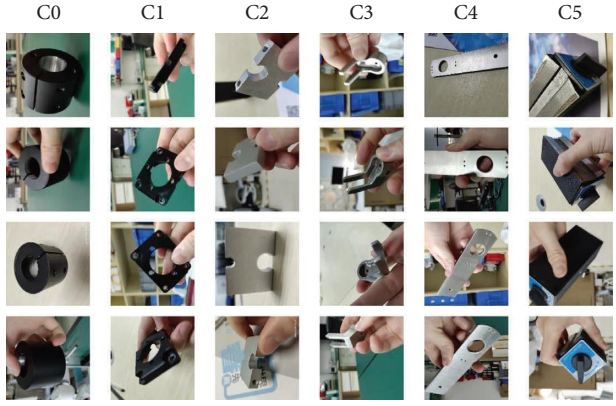

Figure 6: Samples of components of the dataset.

L are more macro, and the detailed decomposition is misty. While the feature maps extracted by M are more detailed, two components of feature maps are blended and extracted to form new, more simplified feature maps in Transform, and each vector represents a feature of the original image. Finally, they are input into the classifier for classification recognition.

5.2. Detail Structure of PLDN. The RGB three-channel image with a 224×224 resolution is used as the standard PDLN input format in the experiment. As shown in Figure 9, once the image is normalized and preprocessed, it is passed to one feature extraction chain with five convolutional blocks and the other with three convolutional blocks. The network performs layer-by-layer convolution operations on the

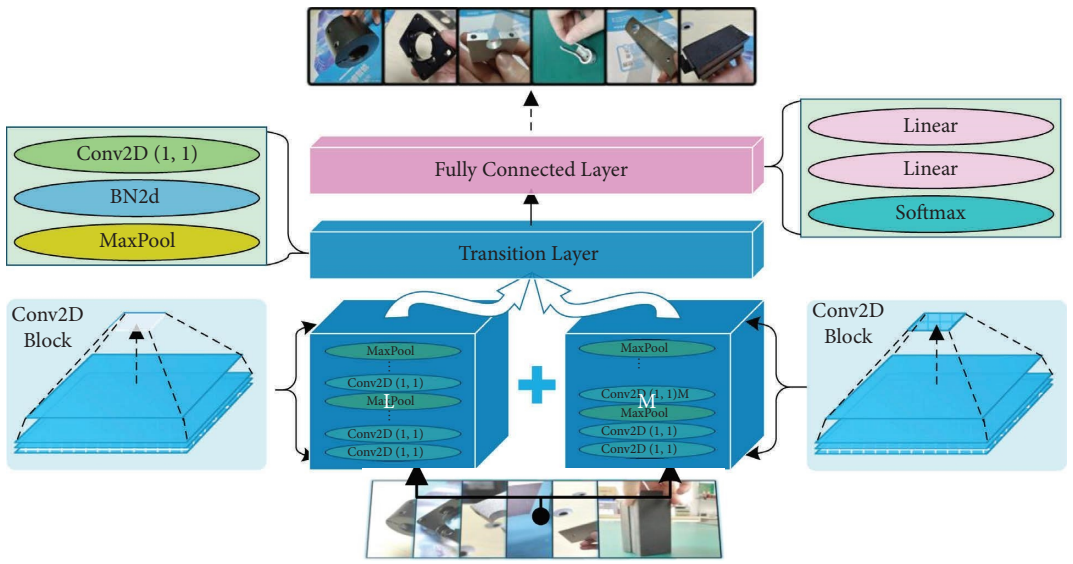

Figure 7: Parallel deep learning network architecture.

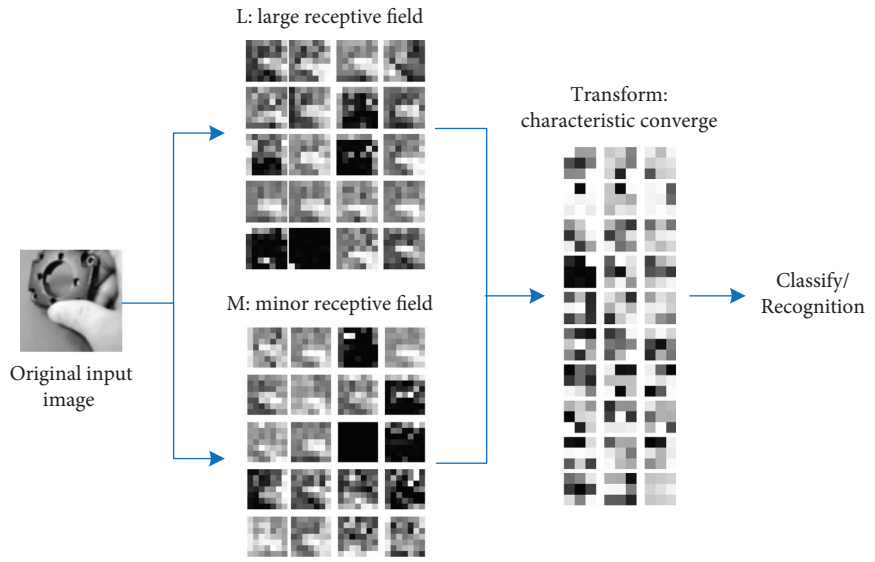

Figure 8: Display of PDLN's image feature extraction and selection process.

image to obtain the feature information. The transition layer then combines and optimizes the two links' (L's and M's) feature quantities. Finally, a classifier with two fully connected layers obtains a final output. To prevent the network from excessive feature extraction and avoid the overfitting phenomenon, the BatchNorm layer and the Dropout layer are added to the network. Additionally, regularization factors are introduced into the optimizer, and the "$L2$" normalization is applied to control each layer's output in the training process.

We consider a simplified model where each convolution block input matrix is denoted as x_i, and the output matrix is y_i. Then,

$$y_i = ReLU(\sigma_i) = ReLU(w_i.x_i + b_i), \tag{7}$$

where w_i and b_i denote convolution block i's weight and bias matrix, respectively. $ReLU$ is the rectified linear activation function. Now, the final network output is

$$\begin{cases} \text{F.C.Layer: } y_{\text{out}} = ReLU(z_f) = ReLU(w_f.x_f + b_f), \\ \text{Tr.Layer: } y_t = x_f = ReLU(z_t) = ReLU(w_t.x_t + b_t), \\ x_t = y_{L_3} + y_{M_5}. \end{cases} \tag{8}$$

The neural network updates the weights by calculating the gradient change rate: $w_i' = w_i - \eta \times \partial y_{\text{out}}/\partial w_i$, $b_i' = b_i - \eta \times \partial y_{\text{out}}/\partial b_i$. Therefore, for the transition layer, the weight update is

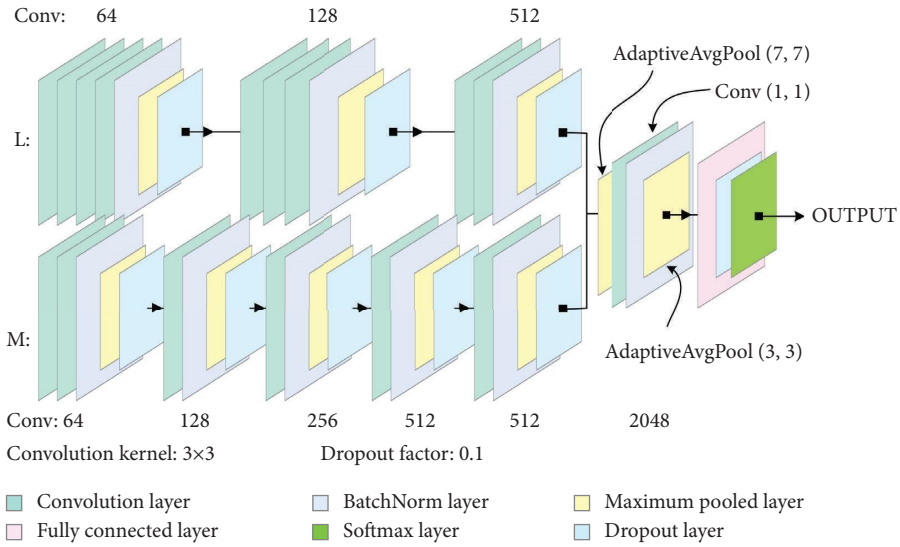

Figure 9: Detailed structure of PDLN.

$$\begin{aligned}\Delta w_t &= \frac{\partial y_{\text{out}}}{\partial w_t} = \frac{\partial y_{\text{out}}}{\partial z_f}.\frac{\partial z_f}{\partial x_t}.\frac{\partial x_f}{\partial y_t}.\frac{\partial y_t}{\partial z_t}.\frac{\partial z_t}{\partial w_t} \\ &= ReLU^{'}\left(z_f\right)w_f.ReLU^{'}\left(z_t\right)x_t \\ &= ReLU^{'}\left(z_f\right)w_f.ReLU^{'}\left(z_t\right).\left(y_{L_3} + y_{M_5}\right).\end{aligned} \tag{9}$$

Following (9), the transition layer weight update Δw_t is related to $(y_{L_3} + y_{M_5})$. Link L is added to the network to increase the variable value x_t, thereby increasing the gradient change rate and effectively alleviating the gradient descent problem.

Multiple adjacent convolutional superpositions can increase the convolution kernel size. This method generates fewer network parameters than the convolutional layer with a convolution kernel of similar size. L and M links relate to different perception fields to capture the overall and detailed image characteristics. Using their combination $(x_t = y_{L_3} + y_{M_5})$, the network can obtain more feature information and find suitable gradients for weight updates while also alleviating the insufficiencies caused by excessive network depth and an inadequate dataset size.

6. Experiments

6.1. Dataset

6.1.1. DATA_ORI. This DATA_ORI dataset consists of rectangular color images originally captured with 224×398 pixels. There are six categories of component images, 340 images in each category, and 2,040 images in total. The dataset is divided into the training data and test sets by 0.85/0.15. Then the training data is divided into the training set and validation set by 9/1. Training and test data used a standard data augmentation scheme (mirroring/rotation). During preprocessing, we compressed the image size to 224×224 and normalized the data using channel mean and standard deviation.

6.1.2. DATA_CON. DATA_CON is generated from DATA_ORI after processing by Algorithm 1. The DATA_ORI size is firstly changed to twice the size, and then Algorithm 1 is used. We choose a value between 1 and 3 for k and try several times (at $k = 3$, i.e., the cropped feature area is 3 times larger, the cropped area already covers the original image completely) to find the most suitable value of k so that the cropped area just covers the object without being too large. The final choice is $k = 1.6$. The square image obtained by Algorithm 1 will be more concentrated on the object (components). The final image size is converted to 224×224 to form DATA_CON. Data enhancement and preprocessing are the same as DATA_ORI.

6.1.3. DATA_DEVICE. DATA_DEVICE is generated by another capturing device (AR) taking pictures. The image is a 224×398 color rectangular image, containing 185 images for six types of components. It is mainly used as a test set to evaluate models. The image size is compressed to 224×224, and the preprocessing is the same as DATA_ORI. A component of the image is shown in Figure 2. The size of the objects in the image is irregular, and the sharpness varies greatly. All of these pose a huge challenge to PDLN-based algorithmic recognition.

6.2. Training. We tested classic deep learning networks such as ResNet18, ResNet152, DenseNet121, DenseNet201, VGG11, and VGG19 and obtained baselines in ResNet and DenseNet according to the official training method. But in VGG, the official training method in [4] has serious overfitting, so we choose to adjust some parameters to adapt to the current training task. In VGG11, the mini-batch is modified to 128 and the baseline is obtained. In VGG19, the mini-batch needs to be modified to 128 and the learning rate modified to 0.001 to obtain the baseline.

In addition, experiments test recent state-of-the-art (SOTA) models on image classification tasks. Big Transfer

(BiT) [24], Convolutional vision Transformer (CvT) [25], and Vision Transformer (ViT) [26] are the models that have performed best in image classification and recognition tasks recently. This study uses their official code and training method to obtain baselines and compare them with PDLN.

The PDLN was trained using stochastic gradient descent (SGD) [27]. The mini-batch size was 128, momentum was 0.9. The training was regularized by weight decay (the "$L2$" penalty multiplier set to 0.0005). The learning rate was initially set to 0.01 and then decreased by a factor of 10 when the validation set accuracy stopped improving. We first used DATA_ORI and DATA_CON for training and testing and then used DATA_DEVICE to evaluate the model's performance. All training and testing were performed on a personal computer (PC). The configuration of the PC is i7-10500, 16G ROM, and RTX3080 12G.

6.3. Testing. We have completed the training of each network in the way mentioned above and obtained baselines. DATA_ORI and DATA_CON are used for training and testing the network, and finally, the obtained model is evaluated with DATA_DEVICE. The results of each baseline are shown in Table 2. We found that the deep neural networks trained and tested on DATA_ORI have a little higher accuracy but a fairly bigger loss than the ones on DATA_CON. However, the deep neural networks trained on DATA_ORI have lower accuracy than the ones on DATA_CON when evaluated on DATA_DEVICE. It indicates that the model's generalization ability is insufficient. Analyzing the performance curves of the test set of the training process of all networks (e.g., vgg11 in Figure 10), it was noticed that almost all of the loss curves showed an increase, and the loss values after stabilization were relatively high, which implies an overfitting problem. In addition, shallower networks tend to have better performance than deeper ones. We suspect that the features available are very limited when the number of learning samples is small, causing the deeper network to overlearn on small data sets and incorrectly use noisy variables such as background as classification criteria, so we need to build a low-depth network. In Table 2, ResNet's 18-layer network (ResNet18) performs very well, and DenseNet performs better than VGG. It considers that the residual network structure has great potential for small datasets. Considering the image characteristics of industrial components, we design a network with residual architecture, which has a large receptive field for extracting global features of the image and also a minor receptive field for extracting local features of the image, and finally, two links are formed (L and M).

6.3.1. Novelty Comparison. For comparison with SOTA models, this study trains and tests BiT, CvT, ViT, and PDLN on our datasets. As results are shown in Table 3, the accuracy of BiT, CvT, and ViT on the test sets of DATA_ORI and DATA_CON is between 70% and 93%, which is far lower than that of PDLN. What is more, the accuracy of PDLN is much higher than other networks on DATA_DEVICE, proving that PDLN has a stronger generalization ability. In addition, the changes in loss and accuracy of PDLN during the training process, which do not fluctuate sharply, are shown in Figure 11. Meaning that there is no overfitting problem in PDLN, and the learning process and training hyperparameter configuration are properly set. Thus, PDLN is advanced and has more potential than the above-mentioned networks on small datasets.

Although the network training accuracies on both DATA_ORI and DATA_CON are not very good, the training process on DATA_CON is better than that on DATA_ORI. Thus, we choose to use DATA_CON to complete the later experiments.

6.3.2. Comparison of Coarse-Fine-Grained (L and M) Network Feature Learning Capability. To explore the feature learning ability of L and M in PDLN and to analyze their impact on the final performance, ablation experiments are established in this study. First, the experiment tests the feature extraction network using only the M part, named PDLN_M. Then, it tests the feature extraction network that only uses the L part, named PDLN_L. The test performances are given in Table 4, where PDLN_M has higher accuracy than PLDN_L, and both PDLN_M and PLDN_L are lower than PDLN. The experiment suggests that when PDLN_M adds a larger receptive field feature (L), i.e., PDLN, not only the network performance is improved but also its adaptability to the heterogeneous source dataset (DATA_DEVICE) is improved. To further verify the correctness of this theory, the experiment visualized the Region of Interest (ROI) in the image at the end of L, M, and the transition layer. As shown in Figure 12, L is more inclined to large regional features, while M is more inclined to micro ones. Under the fusion of the two, the ROI of the transition layer can more accurately capture the key areas of the image. Thus, PDLN achieves higher accuracy and better generalization ability than PDLN_L and PDLN_M.

6.3.3. Parameter Selection for Algorithm 1. The k parameter selection for Algorithm 1 can be used with different values depending on the application scenario. In this study, the control variable method is used to test the effects of different k on the performance of PDLN. As test results are shown in Table 5, the network gets the highest accuracy as 92.43% when $k = 1.2$. And when k is less than 1.2, the performance gets worse as k decreases because the cropped image loses some feature regions. While the cropped image will become large when k is larger than 1.2, causing the proportion of feature regions to decrease, and the performance becomes worse as k increases. Thus, $k = 1.2$ is used for testing each network in this study. When $k = 1.2$, the enhancement results of Algorithm 1 on PDLN are shown in Table 6, which shows that most of the originally incorrectly recognized images can be accurately recognized with the help of Algorithm 1, proving that Algorithm 1 is effective.

By comparing Tables 3 with 2, it can be seen that Algorithm 1 plays a positive role in improving the recognition accuracy of AR images, and can accurately locate the feature concentration area of the image, and the accuracy of PDLN

Table 2: Performance comparison of classical networks.

Model	Training	Testing		Evaluation		Evaluation with Algorithm 1	
	Dataset	Loss	Acc. (%)	Loss	Acc. (%)	Loss	Acc. (%)
VGG11	DATA_ORI	1.2754	82.03	7.4869	23.78	6.9467	36.21
	DATA_CON	1.0606	81.38	4.2180	**33.51**	3.933	51.35
VGG19	DATA_ORI	1.0462	83.66	7.5684	28.10	7.086	32.29
	DATA_CON	0.8374	66.67	**2.3672**	21.08	**1.7511**	34.59
ResNet18	DATA_ORI	0.3248	**93.31**	8.7534	31.35	7.2648	42.70
	DATA_CON	**0.2903**	91.19	13.6997	24.86	5.661	**52.43**
ResNet152	DATA_ORI	1.4845	71.71	7.8313	29.18	7.2735	30.01
	DATA_CON	1.5444	72.28	277.2992	19.45	142.59	28.11
DenseNet121	DATA_ORI	0.6082	85.33	3.9679	24.32	3.7571	30.19
	DATA_CON	0.568	85.48	14.5865	20.54	10.485	34.05
DenseNet201	DATA_ORI	0.5374	87.55	9.1101	32.43	8.1152	43.78
	DATA_CON	0.3379	77.69	11.0234	25.41	6.0151	47.03

The best number performed in these comparisons tested.

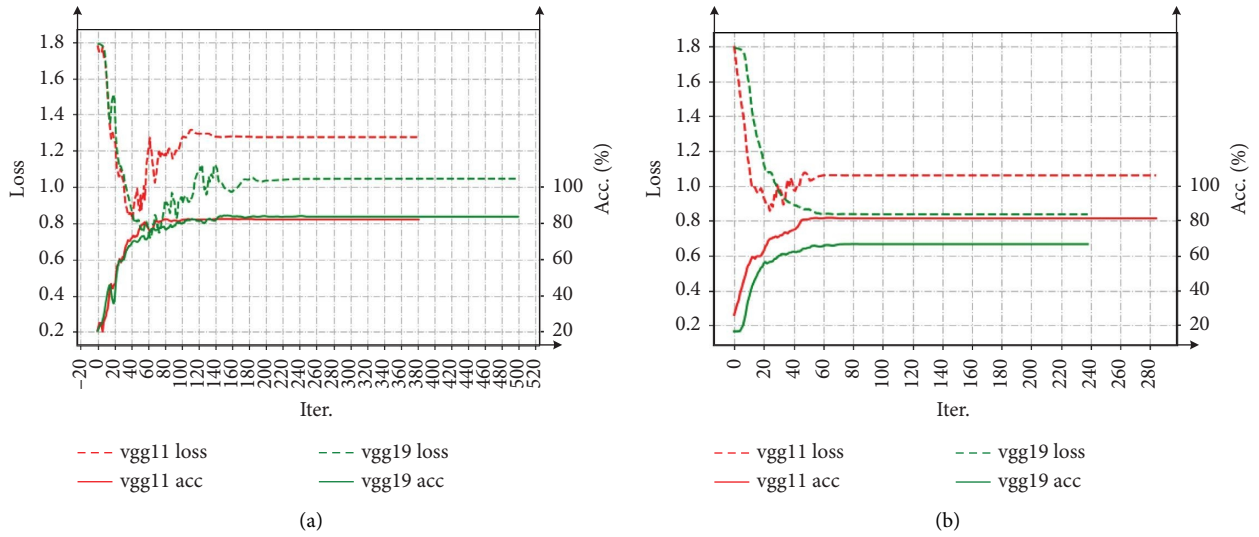

Figure 10: Performance of VGG on different training sets. (a) Train on DATA_ORI. (b) Train on DATA_CON.

Table 3: Performance comparison of SOTA models.

Model	Training	Testing		Evaluation		Evaluation with Algorithm 1	
	Dataset	Loss	Acc. (%)	Loss	Acc. (%)	Loss	Acc. (%)
BiT	DATA_ORI	0.1001	92.15	2.6926	30.23	4.9752	33.51
	DATA_CON	0.2318	90.84	3.3496	34.59	2.4041	47.03
CvT	DATA_ORI	1.0438	83.33	4.7235	24.05	3.4118	44.05
	DATA_CON	0.9596	84.96	3.2217	33.24	2.9417	49.19
ViT	DATA_ORI	3.4291	77.43	13.7787	31.26	9.5347	36.54
	DATA_CON	2.1206	85.07	12.8839	34.37	8.5453	40.62
PDLN	DATA_ORI	0.1303	94.38	2.7066	50.86	3.1891	54.05
	DATA_CON	0.1945	95.11	1.7629	61.62	0.2841	92.43

is higher than that of several networks tested so far, which confirms that PDLN has better learning ability and generalization ability on the dataset of small industrial components. Also, as shown in Tables 5 and 6, intelligent cropping of image feature areas (Algorithm 1) in model evaluation can generally increase the accuracy of the model. Thus, PDLN demonstrates a stronger generalization ability than other networks on a heterogeneous dataset.

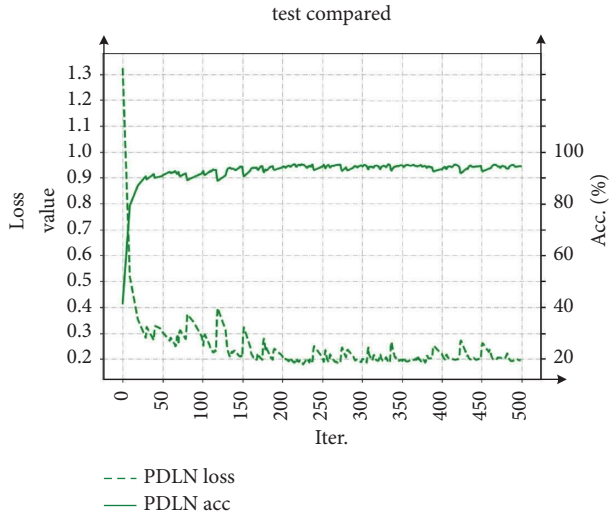

Figure 11: Performance curve of PDLN on DATA_CON.

Table 4: Ablation experiments of PDLN's *L* and *M*.

Model	Training	Testing		Evaluation		Evaluation with Algorithm 1	
	Dataset	Loss	Acc. (%)	Loss	Acc. (%)	Loss	Acc. (%)
PDLN_M	DATA_CON	0.0668	94.37	3.59709	55.14	1.976032	66.49
PDLN_L	DATA_CON	0.3581	92.81	5.9381	46.49	5.584	54.05
PDLN	DATA_CON	0.1945	95.11	1.7629	61.62	0.2841	92.43

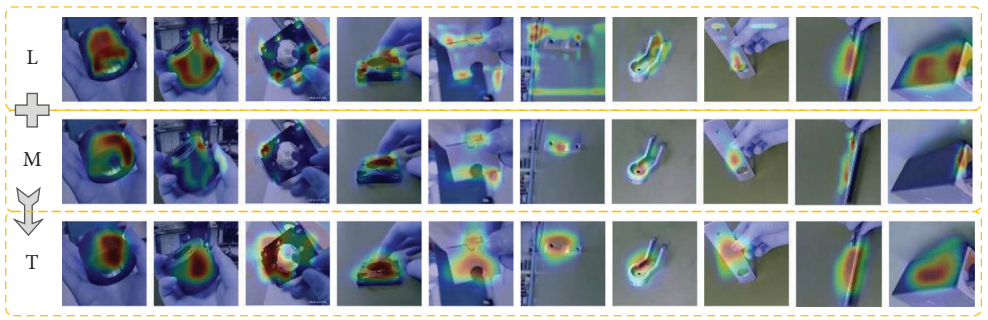

Figure 12: Visualization (heat map) for image regions of interest in the last layer of PDLN's *L*, *M*, and transition layer.

Table 5: The effect of different *k* for improving the recognition accuracy of PDLN on DATA_DEVICE.

False ⟶ true: recognition is wrong without Algorithm 1, and recognition is correct with Algorithm 1									
True ⟶ false: recognition is correct without Algorithm 1, and recognition is wrong with Algorithm 1									
K	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	2.0
False ⟶ true (%)	32.05	33.84	33.51	33.04	31.53	28.96	29.53	28.01	23.54
True ⟶ false (%)	17.02	12.27	2.70	8.19	5.72	4.62	4.24	3.83	3.31
Final Acc. (%)	71.35	76.76	**92.43**	78.92	78.46	78.38	78.92	77.84	73.51

The number that performed best in the multiple comparisons tested.

Table 6: Details of the accuracy improvement for PDLN on DATA_DEVICE by Algorithm 1 when $k = 1.2$.

False ⟶ true: recognition is wrong without Algorithm 1, and recognition is correct with Algorithm 1							
True ⟶ false: recognition is correct without Algorithm 1, and recognition is wrong with Algorithm 1							
Components	C0	C1	C2	C3	C4	C5	Average
False ⟶ true	52.94	54.55	0	25.00	16.13	39.47	33.51
True ⟶ false	0	0	12.00	8.33	0	0	2.70

7. Discussion

Seven deep neural networks, including PDLN and other deep learning networks, are trained on a dataset with 340 sample images per category and less than 2,500 images overall in this section. This study compares the performance of the networks under different data processing methods. This comparison supports the discussion on potential methods to improve the performance of neural networks for industrial component recognition. The findings can be summarized as follows:

(1) In the dataset of industrial components, the background of the images can be complex and unhelpful. Using technical processing to concentrate the image information more on the objects, as Algorithm 1, can effectively improve the recognition of the model and increase the recognition accuracy. In addition, by using these images for training, the generalization ability of the model will be better.

(2) In industrial components recognition, the high accuracy of networks by deepening or decreasing the depth of networks is not reliable because of limited samples. Instead, the proposed coarse-fine-grained feature fusion methodology, which enables the network to consider both global and local features of the image, is a good way to improve the model's ability for distinguishing components in the limited datasets.

(3) Experimental data show that the proposed PDLN is more appropriate than SOTA networks for small dataset learning in the field of automotive equipment manufacturing. After 200 training rounds, PDLN achieves a recognition accuracy of approximately 98%. The accuracy of 92% was also maintained in additional datasets, thus proving more robust and accurate than traditional networks.

8. Conclusion

The rapid updates and wide range of products in the automotive manufacturing industry make it difficult to build large datasets. This paper proposes an intelligent recognition method for automotive components based on coarse-fine-grained feature fusion and deep learning from the perspective of enterprise economic efficiency. This method contains an image intelligent cropping algorithm (Algorithm 1) based on feature detection, and through the designed architecture, dataset production, network learning, and model application, it is completed to achieve reliable component recognition accuracy. Experiments demonstrate that the proposed method can obtain good robustness and 95.11% recognition accuracy in learning with limited samples. In addition, the designed Algorithm 1 can automate the generation of datasets and form an automated system for the whole process of "data collection-network learning-model application." Thus, new datasets and available models can be produced promptly in the rapid product update, providing solutions for the intelligence of the manufacturing industry. The future will be based on deep learning to achieve more accurate industrial dataset generation methods and model upgrades and iterations.

Data Availability

Datasets used in this study are available at the Google Cloud: https://drive.google.com/drive/folders/1-wnpD8LuSp6dP5eIO9BGNEahMqjULTKs?usp=sharing.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the Natural Science Foundation of Guangdong Province, China (2021A1515011946), and the Key Program of the National Natural Science Foundation of China (No. U1801264) for the support.

References

[1] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-Driven fault Diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[2] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection for Digital Twin in Smart manufacturing with industrial Cyber-Physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR), Las Vegas, NV, USA, June 2016.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, https://arxiv.org/abs/1409.1556.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature Hierarchies for accurate object detection and Semantic Segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014, https://arxiv.org/abs/1311.2524.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only Look once: Unified, Real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016, https://arxiv.org/abs/1506.02640.

[7] S.-Y. Chen, Y.-C. Cheng, W.-L. Yang, and M. Y. Wang, "Surface defect detection of Wet-Blue Leather using Hyperspectral imaging," *IEEE Access*, vol. 9, pp. 127685–127702, 2021.

[8] K. Wang, R. B. Gopaluni, J. Chen, and Z. Song, "Deep learning of complex batch process data and its application on quality Prediction," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7233–7242, 2020.

[9] X. Bu, J. Peng, J. Yan, T. Tan, and Z. Zhang, "GAIA: A Transfer Learning System of Object Detection that Fits Your Needs," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 274–283, 2021, https://arxiv.org/abs/2106.11346.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin loss for deep Face recognition," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4694, Long Beach, CA, USA, June 2019.

[11] Z. Fan, L. Shi, C. Xi, H. Wang, S. Wang, and G. Wu, "Real time power equipment meter recognition based on deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[12] Y. Chang and W. Wang, "A deep learning-based Weld defect classification method using Radiographic images with a Cylindrical Projection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[13] D. G. Lema, O. D. Pedrayes, R. Usamentiaga, P. Venegas, and D. F. Garcia, "Automated detection of Subsurface defects using active Thermography and deep learning object Detectors," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[14] N. Banus, I. Boada, A. Bardera, and P. Toldra, "A deep-learning based solution to automatically control Closure and seal of Pizza Packages," *IEEE Access*, vol. 9, pp. 167267–167281, 2021.

[15] X. Zhang, C. Wang, T. Wu, and Y. Wang, "Application of intelligent recognition technology in recognition of Mechanical material structure," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 8909122, 7 pages, 2022.

[16] Q. Luo, J. Wang, M. Gao, H Lin, H Zhou, and Q Miao, "G.-Yolox: A lightweight network for detecting vehicle types," *Journal of Sensors*, vol. 2022, Article ID 4488400, 2022.

[17] L. Wu, L. Zhang, J. Shi, Y. Zhang, and J Wan, "Damage detection of grotto murals based on lightweight neural network," *Computers and Electrical Engineering*, vol. 102, Article ID 108237, 2022.

[18] S. Niu, B. Li, X. Wang, and H. Lin, "Defect image sample generation with GAN for improving defect recognition," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2020.

[19] G. Huang, Z. Liu, L. V. D. Maaten, and K Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, July 2017.

[20] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, pp. 839–846, IEEE Cat. No.98CH36271), Bombay, India, January 1998.

[21] K. S. Rani and R. V. S. Satyanarayana, "Image denoising using boundary discriminated switching bilateral filter with highly corrupted universal nois," in *Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, pp. 1515–1521, ICECDS), Chennai, India, Auguest 2017.

[22] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, 2002.

[23] E. Rublee, V. Rabaud, K. Konolige, and G Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, November 2011.

[24] A. Kolesnikov, L. Beyer, X. Zhai et al., "Big Transfer (BiT): General visual Representation learning," *Computer Vision – ECCV*, pp. 491–507, Glasgow, UK, Auguest 2020.

[25] H. Wu, B. Xiao, N. Codella et al., "CvT: Introducing Convolutions to Vision Transformers," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31, Montreal, QC, Canada, October 2021.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is Worth 16x16 Words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, 2021, https://arxiv.org/abs/2010.11929.

[27] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to Handwritten Zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.