Neural Computing and Applications (2021) 33:14413–14428
https://doi.org/10.1007/s00521-021-06081-9

ORIGINAL ARTICLE

Automatic identification of commodity label images using lightweight attention network

Junde Chen[1] · Adnan Zeb[1] · Shuangyuan Yang[1] · Defu Zhang[1] · Y. A. Nanehkaran[1]

Received: 24 October 2020 / Accepted: 20 April 2021 / Published online: 6 May 2021

Abstract

Recent research has raised interest in applying image classification techniques to automatically identify the commodity label images for the business automation of retail enterprises. These techniques can help enterprises improve their service efficiency and realize digital transformation. In this work, we developed a lightweight attention network with a small size and comparable precision, namely MS-DenseNet, to identify the commodity label images. MS-DenseNet is based on the recent well-known DenseNet architecture, where we replaced the regular planner convolution in dense blocks with depthwise separable convolution to compress the model size. Further, the SE modules were incorporated in the proposed network to highlight the useful feature channels while suppressing the useless feature channels, which made good use of interdependencies between channels and realized the maximum reuse of inter-channel relations. Besides, the two-stage progressive strategy was adopted in model training. The proposed procedure achieved significant performance gain with an average accuracy of 97.60% on the identification of commodity label images task. Also, it realized a 94.90% average accuracy on public datasets. The experimental findings present a substantial performance compared with existing methods and also demonstrate the effectiveness and extensibility of the proposed procedure. Our code is available at https://github.com/xtu502/Automatic-identification-of-commodity-label-images.

Keywords Image recognition · Image classification · CNNs · MS-DenseNet · Lightweight network

1 Introduction

Corporations are facing various challenges in the era of the digital economy, which potentially impacts the whole corporation. More and more corporations have paid attention to digital transformation and business automation, which are becoming the primary concerns and emerging subjects in the process of information construction. Particularly, simple, repetitive businesses can be firstly realized automation in practical applications. For retail enterprises, automatically identifying commodities through the captured digital images may prove benefits in locating and finding the right objects timely in large fields of goods. Therefore, seeking a simple, quick, low-cost, and reliable tool to automatically identify the commodity label images is of great practical importance.

With rapid economic development and technological progress, machine learning (ML) approaches are emphasized by both academic researchers and industrial practitioners due to their ability of automating work for improving efficiency. Therefore, ML, along with image processing, has gained considerable interest in recent years [1–3]. Subsequently, many recent works have considered image recognition aiming at diverse product images, and various classifiers are used to distinguish between image labels. Recently, automatic image identification techniques have attracted great interest due to significant

✉ Shuangyuan Yang
yangshuangyuan@xmu.edu.cn

✉ Defu Zhang
dfzhang@xmu.edu.cn

Junde Chen
chen2wo@126.com

Adnan Zeb
adnanzeb@stu.xmu.edu.cn

Y. A. Nanehkaran
artavil20@gmail.com

[1] School of Informatics, Xiamen University, Xiamen 361005, China

improvements in digital cameras and computational resources. These techniques have been effectively utilized in many domains such as food analysis [4, 5], plant disease identification [6, 7], health care [8, 9], biometrics [10], among others [11–13], and have achieved impressive performance. For instance, a method that utilized SVM and TF-IDF models to recognize commodity images was proposed by Ma et al. [14]. However, their method is based on manual extraction of the features of the pre-processed corpus, which requires enough time, particularly, when the dataset contains a large number of images. Apart from that, six ML algorithms, including artificial neural network (ANN) and SVM were adopted to categorize different Pedestrians' events based on global positioning system (GPS) and inertial measurement unit (IMU) signals [15]. In terms of average accuracy, extra tree (ET), with an accuracy of 91%, has been found to be the best ML model among the existing ones. In another research, a method was put forward for the goods detection and identification using the k-means clustering along with SVM classification algorithms [16]. This procedure integrates classical HOG detection with the SIFT feature-based bag of words model to identity various products. As mentioned above, machine-learning-based image identification typically involves feature extraction and image classification processes in which feature extraction is essential, and its quality significantly affects the end result of image identification. However, the traditional ML methods employ hand-crafted features to represent the underlying characteristics of images. Such techniques capture certain morphological attributes (color, texture, etc.) and require expensive work and expert knowledge. Although these traditional ML algorithms have a good predictive performance on small data samples, their efficiency is declined when manually extracting multiple features. Moreover, the approach of manually extracting features is subjective and sensitive to image noise and complex background conditions too. Recently, deep learning (DL), a sub-category of ML, has been proposed to address most technical challenges. Particularly, the convolutional neural networks (CNNs) can automatically extract image features without human intervention and are quickly becoming the leading methods for image recognition and classification [17–19]. Zou et al. [20] trained a cascaded CNN on a large commodity images dataset, and they achieved the highest performance of mean average precision (mAP) at 94.6%. However, because it is a deep CNN model, the size of the model is large and it is not suitable for deployment in mobile or portable device applications. In another work, a deep learning-based system was put forward by Chen et al. [21] for the commodity image retrieval, and their system reached 70% and 39% accuracy and recall, respectively. This model still has room for improvement of the accuracy. Additionally, motivated by the ability to transfer parameters using transfer learning, Cao et al. [22] proposed a novel CNN-based method for the two-attribute e-commerce image classification, and they achieved the test accuracy of 95.2% for type and 94.3% for color, respectively. As mentioned earlier, though reasonable promising findings have been reported in the literature, there are many parameters needed to be trained for the deep CNNs with large volumes, which requires high memory storage.

Moreover, the collection of large-scale data samples is typically complex and challenging, while reducing the number of parameters results in lowering the learning capability of the CNN models. Therefore, there is a trade-off between the computational resources and classification accuracy in CNN models, and the recent research and application of lightweight deep CNNs have gained significant attention. In this paper, considering the computing power and memory resources, we evaluate the performance of the currently popular lightweight neural networks for the identification of commodity label images. We also propose MS-DenseNet, a novel neural architecture with limited model size and high classification accuracy, compared to existing methods. Overall, our main contributions are summarized as follows.

1. We have collected a commodity label image dataset from real-life business scenarios. It contains approximately 1,000 images captured for physical commodity labels, and each image has been assigned to a specific category. We expect this dataset to encourage further studies on commodity label identification.
2. We have used DenseNet as the backbone architecture. In DenseNet, we reserved the composition structure of the transition layer and replaced the standard convolution with depthwise separable convolution in the dense blocks to reduce computational overhead.
3. The SE block was incorporated in the network to highlight the useful feature channels while suppressing the unwanted feature channels, which makes good use of channel-wise attention to learn the significance of the interdependencies between channels and realize the maximum reuse of inter-channel relation.
4. The two-stage progressive strategy was adopted in model training, and the loss function was optimized. To keep more focus on positive samples and address multi-classification tasks, we enhanced the Focal-Loss function and substituted for the classical Cross-Entropy Loss function.

The rest of this writing is presented as follows: Sect. 2 provides details of the collected image dataset and an overall flow introduction. It also describes methods for performing the identification of commodity label images, relevant existing work, and the proposed method.

Subsequently, Sect. 3 presents a number of experiments to evaluate the model performance and compares the results with various existing methods. Finally, Sect. 4 summarizes this writing and presents solid suggestions for further work.

2 Materials and methods

2.1 Data collection

Approximately 1,000 sample images captured from physical commodity labels were provided by Pzcnet Ltd., Xiamen, China (http://www.pzcnet.com/). The images are originally colored images of inconsistent sizes, and a consumer-level color camera with the type of Nikon Coolpix S3100 is used to photograph these images. The maximum pixels of this CCD camera are 4320 × 3240. These images were photographed in real business events with varied backdrops and lighting intensities. It is worth noting that all images were taken without a flashlight and digital zoom during daylight from 9:00 to 17:00 on Aug 29, 2019. In particular, some of these images are similar, however, captured with different angles and distances. All the images were carefully labeled and stored in JPG format. There are a total of 45 categories in these samples, and each category contains an unequal number of images representing the same sample. Further, to fit the model, the Photoshop tool was used to transform these images into the RGB model, and their dimensions were resized to 224 × 224 pixels. Furthermore, the categorical variable was encoded into a one-hot vector for model training. A few sample images are shown in Fig. 1.

2.2 Overall process

A brief overview of the proposed procedure for the identification of commodity label images is presented in the following. Initially, the collected images are labeled according to the expertize of the field experts, and image processing techniques, including image sharpening, resizing, and edge filling, are applied to all the collected images. Then, in addition to a fixed number of original images allocated for model validation, the sample images are augmented using the data enhancement scheme. Including random translation, flipping, and scaling, multiple geometric transformations are implemented to produce new diverse synthetic images for enriching the dataset. Next, the sample images are given to the proposed lightweight MS-DenseNet, which preserves the composition and structure of transition layers for network training. Finally, the generated optimal model is then used to predict the class of commodity label images, thereby obtaining the final identification results. Figure 2 shows an overall flowchart of commodity label image identification, and the complementary details of each given process are included in the following sections.

2.3 Related work

2.3.1 DenseNet

DenseNet [23] is a popular deep learning model comprised of dense blocks and transition layers. It solves the vanishing gradient problem, enhances the propagation of features, and reduces parameters for the deep networks. It is characterized that each layer in the dense block directly connects to its front layers for feature reuse. Dense blocks

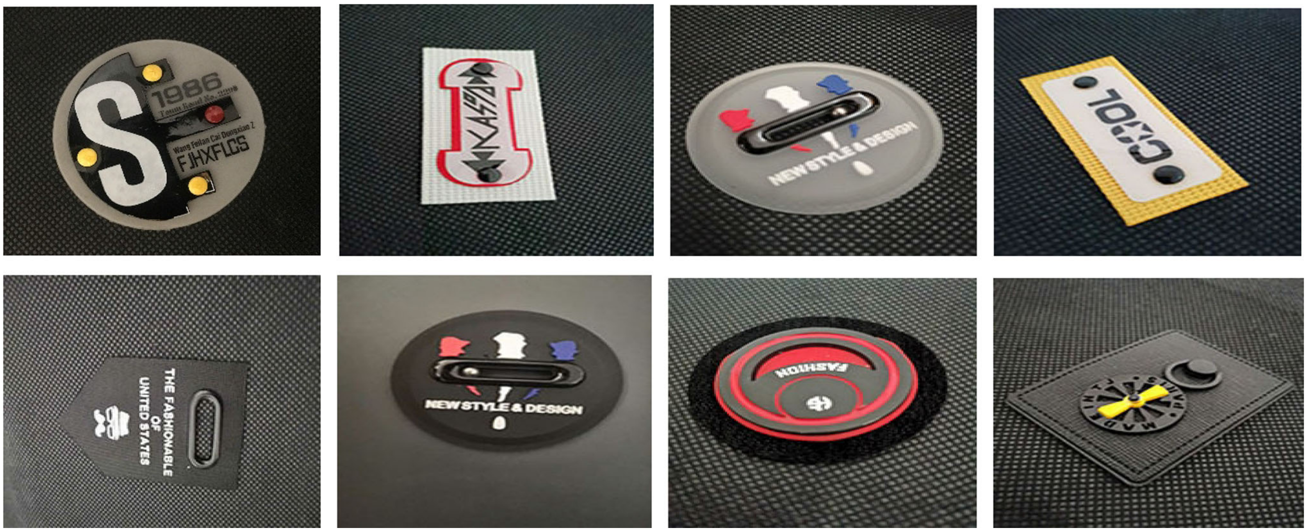


Fig. 1 Examples of commodity label images

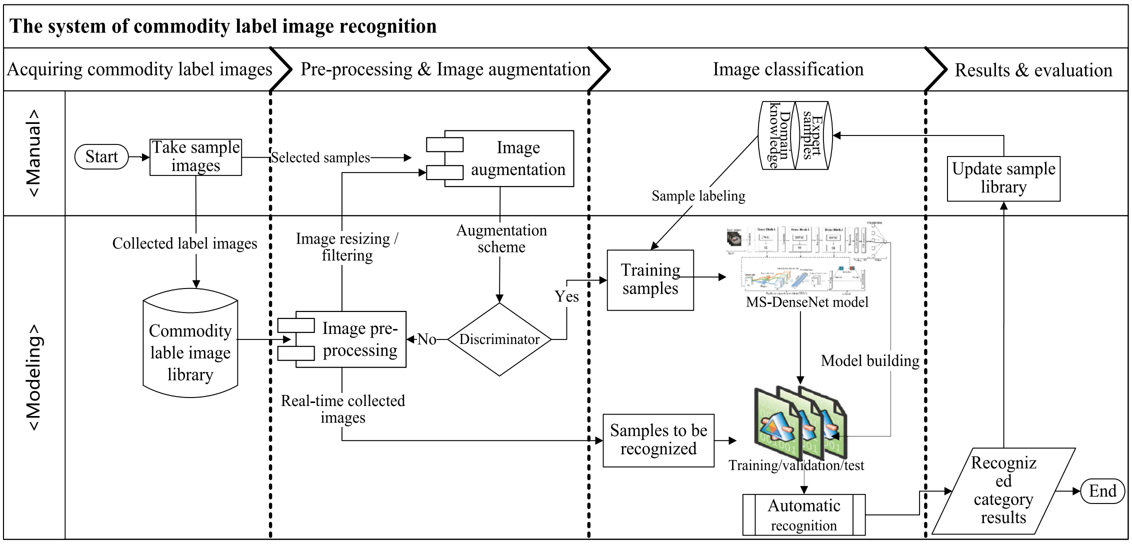


Fig. 2 An overall flowchart of commodity label image recognition

contain a fixed number of repeated Batch Normalization, ReLU activation function, 3 × 3 and 1 × 1 convolution sets, as shown by an example of a 3-layered dense block in Fig. 3. Each layer receives feature maps from all previous layers and generates for the latter layers, as defined in the following Eq. (1).

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \tag{1}$$

where l is the number of CNN layers, $[x_0, x_1,\ldots, x_{l-1}]$ denotes cascading the feature maps from 0 to $l-1$ layers as subsequent layer's input. Then, to decrease the number of input feature maps, the transition layer joins two neighboring dense blocks and decreases the dimension as a 1 × 1 convolution pooling layer. A typical transition layer composition has a BN-ReLU-Conv loop connection, consisting of a batch normalization, a ReLU activation function, and a 1 × 1 convolution operation, followed by a 2 × 2 average pooling layer for reducing dimensionality. As stated previously, DenseNet avoids gradient vanishing problem and allows feature reuse. Although it exhibits competitive advantages, DenseNet also has obvious shortcomings. First, each layer is connected to all other layers, where such dense connections increase the amount of calculation and consume more memories. Second, each layer typically concatenates feature maps obtained from the previous layers without considering the channel relation features, which does not involve the interdependencies between channels. On the basis of the above discussion, we believe that by compacting the model size, limiting the number of parameters, modeling the channel relation features, and performing the channel-wise feature recalibration, the network performance can be further enhanced. In particular, each convolutional layer is relatively narrow such as 1 × 1 and 3 × 3. Thus, a few feature maps are learned to minimize redundancy. For this, the article reserves the transition layer's structure and composition, as further discussed in Sect. 2.4.

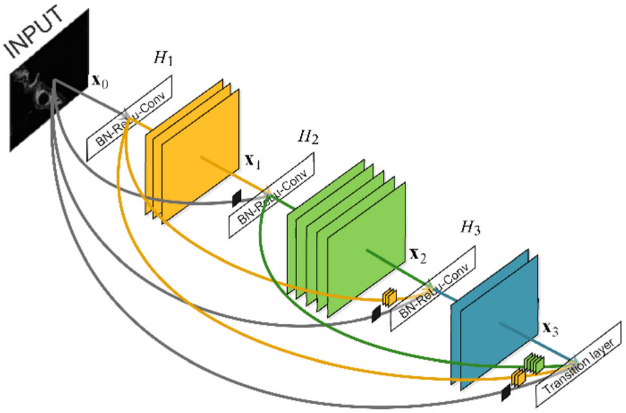


Fig. 3 The schematic diagram of a 3-layer dense block [23]

2.3.2 MobileNet

MobileNet [24] is a mobile-first computer vision model designed to effectively maximize accuracy while considering the resource limitations for deploying a deep learning application. MobileNet model is small, lightweight, low-latency, and less complex compared to other state-of-art models. It has attained impressive performance on multiple image classification and identification tasks. MobileNet uses a depth-wise separable convolution (DWSC)-based network architecture, which extends the regular planner convolution into two stages: a depth-wise convolution (DWC) and a point-wise convolution (PWC) [25]. Generally, regular convolution (RC) [26] slides over the image

(y) through a filter containing weights (W), as calculated by:

$$\mathrm{RC}(W, y)_{(i,j)} = \sum_{h=0}^{H} \sum_{g=0}^{G} \sum_{m=0}^{M} W_{(h,g,m)} \times y_{(i+h,j+g,m)} \tag{2}$$

where H and G represent the height and width of the image, respectively. M is the number of filters, and W indicates the weights of filters. In contrast, the DWSC is typically implemented as follows:

$$\mathrm{DWSC}(W_d, W_p, y)_{(i,j)} = \mathrm{PWC}_{(i,j)}(W_p, \mathrm{DWC}(W_d, y)_{(i,j)}) \tag{3}$$

Here, in DWC, each channel is conducted the convolutional operation with one filter for the input map, and the PWC adopts the results of DWC to implement a 1 × 1 convolution kernel operation. With this method, the final output of DWSC can be obtained accordingly, and the detailed operations of DWC and PWC are separately expressed by:

$$\mathrm{DWC}(W_d, y)_{(i,j)} = \sum_{h=0}^{H} \sum_{g=0}^{G} W_{d(h,g)} \odot y_{(i+h,j+g)} \tag{4}$$

$$\mathrm{PWC}(W_p, y)_{(i,j)} = \sum_{m=0}^{M} W_m \times y_{(i,j,m)} \tag{5}$$

where y denotes the input image and (i, j) the index position of the image. Figure 4 presents the comparison between regular convolution and depthwise separable convolution.

2.3.3 SE block

The key component of SENet is the Squeeze-and-Excitation (SE) [27] block, which aims to improve performance by directly modeling the inter-dependencies between the feature channels. SE block comprises two main operations: squeeze and excitation. The squeeze operation shrinks feature maps $\in \mathrm{R}^{w \times h \times c2}$ through spatial dimensions ($w \times h$), which is obtained through global average pooling to compute a channel-wise statistic z, as follows:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{6}$$

where z_c is the c-th statistic value, u_c represents the c-th feature map of previous convolution operation, H and W represent the height and width of u_c, separately. The excitation operation extracts the dependencies of channels entirely using information aggregated during squeeze operation, as calculated in Eq. (7), where a basic gating function with sigmoid activation is incorporated.

$$s = F_{\mathrm{ex}}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{7}$$

Here δ refers to the ReLU [28] function, $W_1 \in \mathrm{R}^{C \times c/r}$, and $W_2 \in \mathrm{R}^{C \times c/r}$, r is a constant-initialized reduction ratio. In particular, two fully connected (FC) layers around nonlinearity learns the parameters W_1 and W_2. Thus, the output of the SE block is generated by rescaling u_c with the activations s:

$$\tilde{x}_c = F_{\mathrm{scale}}(u_c, s_c) = s_c \cdot u_c \tag{8}$$

where $[\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_c]$ represents the final output $\tilde{X}$. Figure 5 depicts the flowchart structure of the SE block.

As mentioned previously, the dense block is especially narrow, which means that each feature map may have important information about each other. An intuition is whether the correlation between the channels in dense blocks can be fully utilized to improve the model effectiveness. Therefore, in this work, the SE block was added in our network to learn the interdependencies between channels for improving the performance of the model.

2.4 Proposed approach

2.4.1 MS-DenseNet model

As mentioned earlier, DenseNet is an existing state-of-the-art model with excellent feature extraction capabilities as its feature layers obtain all the features from previous layers and provides its feature-maps as input to all subsequent layers. In this manner, DenseNet has attained successful performance gains on multiple classifications and detection tasks. Besides, depthwise separable convolution, which achieves an optimum trade-off between the computational memory and classification accuracy, is the

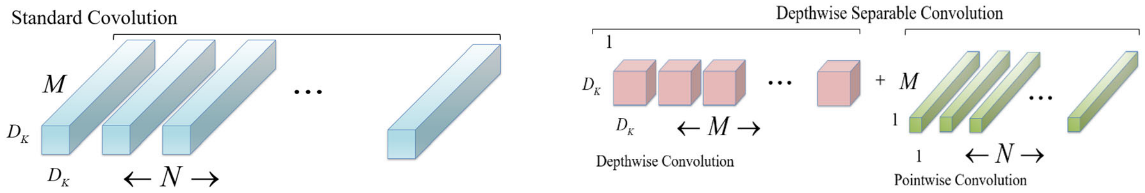

Standard Covolution
M
D_K
D_K
...
$\leftarrow N \rightarrow$
Depthwise Separable Convolution
1
D_K
...
D_K
$\leftarrow M \rightarrow$
Depthwise Convolution
+ M
1
...
1
$\leftarrow N \rightarrow$
Pointwise Convolution

Fig. 4 Regular convolution and depthwise separable convolution

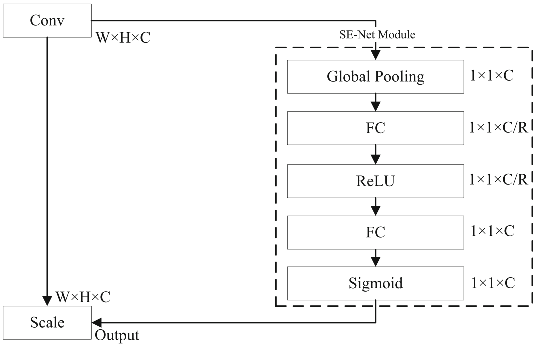


Fig. 5 The SE building block [27]

primary trend of lightweight CNN development. Including the MobileNet, NASNetMobile [29], ShuffleNet [30], and EfficientNet [31], the well-known lightweight CNN models are all using the depthwise separable convolution as the core components. Therefore, for the architecture of DenseNet, we replaced the standard convolution with depthwise separable convolution in the dense blocks and incorporated the SE module in the dense blocks to effectively utilize the channel relationship features. By doing this, a mobile SE-DenseNet, which we termed the MS-DenseNet, was formed and used to identify the commodity label images. More specifically, the proposed architecture is developed using the following processes.

Based on DenseNet's architecture, we preserved the transition layer's composition and structure, which is composed of a batch normalization, a ReLU activation function, and a 1 × 1 convolution followed by a 2 × 2 average pooling layer. Then, the convolution operation in the dense block was replaced by depthwise separable convolution, which aimed to compress the model size and make effective use of model parameters. In other terms, all the dense blocks in the DenseNet model, Dense_Block1 to Dense_Block3, were modified by substituting the depthwise separable convolution for the traditional convolutional layers to minimize model parameters for increasing efficiency of the DenseNet. Note that the size of the convolution kernel used in depthwise separable convolution is 3 × 3. Assuming the size of the input feature map is $D_f \times D_f$, the number of channels is M, the size of the convolutional kernel is $D_k \times D_k$, and the number of output channels is N, then the computation cost of traditional convolution C_1 is equal to $D^{2}{}_{f}MND^{2}{}_{k,}$ and that of the depthwise separable convolution C_2 is $D^{2}{}_{f}M(D^{2}{}_{k} + N)$. In practice, depth-wise separable convolution is commonly implemented with a 3 × 3 convolution with a higher number of output channels N. Therefore, the number of parameters of the proposed approach is around one-ninth of that of the original method. Furthermore, to fully utilize the correlation between the channels and improve the model performance, the SE modules were incorporated in the network to enhance useful features and suppress unwanted features by learning the importance of each channel. Note that instead of the common method of adding the SE module to the convolution layer or the depthwise separable convolution layer, we embedded the SE module to the dense block in order not to add too many parameters. Concretely, the SE module was embedded between the ReLU and convolution layers in each dense block of the proposed network to learn the significance of the interdependency relation between channels and perform dynamic channel-wise feature recalibration. After that, a global pooling layer replaced the fully-connected layer, adding a new fully-connected Softmax layer with the practical number of classes as the network's top classification layer. Figure 6 depicts the proposed network's architecture, and the corresponding network parameters are given in Table 1.

2.4.2 Loss function

In general, the Cross-Entropy Loss function is the commonly used loss function in deep CNN models, and it is defined as follows:

$$L = -\sum_{c=1}^{N} y_c \log(p_c) \tag{9}$$

where N is the number of categories, y_c is the indicator variable ($y \in (0,1)$); if category c is the same as the category of the sample, it is 1; otherwise, 0. p_c is the predicted probability of a particular sample belonging to category c. Since the predicted loss weights of the Cross-Entropy function are regarded as the same for both positive and negative samples, Lin et al. [32] introduced the Focal Loss function, expressed as follows:

$$L(p_c) = -\alpha_c(1 - p_c)^{\gamma}\log(p_c) \tag{10}$$

$$p_c = \begin{cases} p, & y = 1 \\ 1 - p, & y = 0 \end{cases} \tag{11}$$

$$\alpha_c = \begin{cases} \alpha, & y = 1 \\ 1 - \alpha, & y = 0 \end{cases} \tag{12}$$

where p denotes the prediction probability, and a is the weighting factor of the loss function when the corresponding category is 1 (binary issue), and γ is a hyperparameter of modulating factor. It's worth noting that the Focal Loss function is mainly suitable for target detection tasks that require binary classification. The identification of commodity label images, on the other hand, is a multi-classification problem, so the classical Focal Loss function was improved and then used to replace the original Cross-Entropy loss function in the deep CNN, as calculated in Eqs. (13–15).

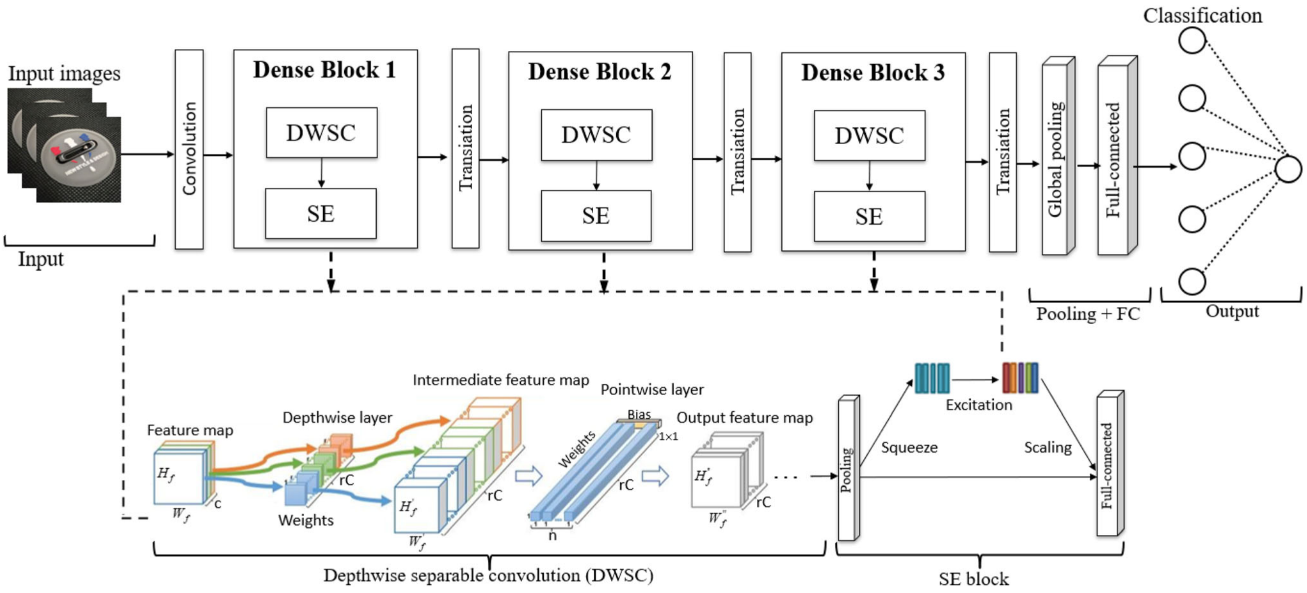


Fig. 6 The proposed network architecture

Table 1 The primary parameters of the MS-DenseNet network

Layer name	Type	Kernel Size	Stride	Output Size	Maps
Input_1	Input Layer	–	–	224 × 224	–
Convolution	Conv2D	7 × 7	2	112 × 112	64
Pooling	MaxPooling2D	3 × 3	2	56 × 56	64
Dense Block1	SeparableConv2D	3 × 3	1	56 × 56	32
SE_Block 1	SeBlock	1 × 1	1	56 × 56	64
Transform1	Conv2D	1 × 1	1	56 × 56	160
Transform1	Average pooling	2 × 2	2	28 × 28	160
Dense Block2	SeparableConv2D	3 × 3	1	28 × 28	32
SE_Block 2	SeBlock	1 × 1	1	28 × 28	80
Transform 2	Conv2D	1 × 1	1	28 × 28	272
Transform 2	Average pooling	2 × 2	2	14 × 14	272
Dense Block3	SeparableConv2D	3 × 3	1	14 × 14	32
SE_Block 3	SeBlock	1 × 1	1	14 × 14	136
Transform 3	Conv2D	1 × 1	1	14 × 14	520
Transform 3	Average pooling	2 × 2	2	7 × 7	520
Convolution	Conv2D	3 × 3	1	7 × 7	32
Global_pool	Global pooling	–	–	1 × 1	516
Dropout	Dropout layer	–	–	1 × 1	516
Classification	Dense (Softmax)	Classifier	–	1 × 1 × 45	45

$$FL_{\mathrm{mult}}(p_c) = -\sum_{c=1}^{N} \alpha_c (1-p_c)^{\gamma} y_c \log(p_c) \tag{13}$$

$$a_c = \mathrm{count}(x_t)/\mathrm{count}(x_t \in c) \tag{14}$$

$$y_c = \begin{cases} 1, & c = \mathrm{true\_class} \\ 0, & c \neq \mathrm{true\_class} \end{cases} \tag{15}$$

where N denotes the total number of categories, c is the class index, p_c denotes the distribution of the prediction probability, and γ is the hyperparameter. Particularly, the weight factor a_c is calculated through dividing the total number of samples by the number of samples in class c, which performs the weighting of the categories and makes the a_c of each class greater than or equal to 1. As a consequence, the initial loss value of the FL_{mult} function is relatively large, which makes the loss value of the model decay quickly to a small value, accelerating the convergence process of the model. In addition, the structure and composition of the DenseNet reserved in the network alleviate the problem of vanishing-gradient [23], which ensures the effective convergence of the model.

2.4.3 Model training

In this study, we trained the model in two phases. In the first phase of training, only feature extractor was trained for some epochs, and this stage primarily performed the coarse-tuning for the network parameters. Then in the second phase, the weights trained in the first phase were injected in, and the network was retrained (fine-tuned) using the target dataset. This kind of progressive training can help the model discover the large-scale structures in commodity label images first, and then shift its focus to detailed features step-by-step, with no need to learn all scales at the same time. More precisely, each individual process for model training is discussed below.

Initially, using the target dataset, the network is learned from scratch, and this process mainly obtains the initial weights of the network. Here, Adam [33] optimizer was used to update the weight.

$$\theta_{c+1} = \theta_c - \eta * \hat{b_c} / \left(\sqrt{\hat{s_c}} + \varepsilon\right) \tag{16}$$

where θ is the weight matrix, c denotes the class index, η represents learning rate, $\hat{b_c}$, and $\hat{s}_c$ denotes bias-corrected first and second moments.

Subsequently, after obtaining the initial weights, the network shifts its attention to delicate details. The weight parameters of the first stage were then injected into network and were retained on the target datasets. Thus, the network weights were fine-tuned based on the initial value rather than inferred from scratch. With this method, the optimum model was obtained for the identification of commodity label images. In this stage, stochastic gradient descent (SGD) [34] was employed as an optimizer to update weights, as defined by

$$\theta_{c+1} = \mu\theta_c - \eta J(\theta_c) \tag{17}$$

$$J(\theta) = \frac{1}{N}\sum_{c=1}^{C} L(f(\theta; x_c), y_c) + \lambda R(\theta) \tag{18}$$

where μ is the momentum weight, $J(\cdot)$ is the average loss, $L(\cdot)$ is the loss function, $f(\cdot)$ is the predicted output of the network with the current weights θ, and λ is the Lagrange multiplier for the weight decay or regularization term R.

3 Experimental results and analysis

Apart from a few image pre-processing algorithms, the experiments were primarily implemented using Anaconda3 (Python 3.6), where the Tensorflow, OpenCV-python3, Keras, and other libraries were utilized and accelerated by GPU. The experimental hardware environments include Intel® Xeon(R) E5-2620 CPU (2.10 GHz), 64 GB RAM, and GeForce RTX 2080 graphics card (CUDA 10.2), which are used for the program running.

3.1 Experiments with label images

A number of experiments were performed on the commodity label images to analyze the performance of the proposed approach. Apart from some original images that were reserved for model validation, the remaining samples were divided into training and test sets in an 8:2 ratio. That is, the training and test sets of sample images were used for training to determine if the model was overfitted, whereas new unseen samples were used to validate the model. It is noteworthy that in this case, “Unseen” images refer to commodity label images that CNN models have not used previously during both training and testing. In addition, we applied the data augmentation scheme to expand the sample images in order to ensure image diversity and prevent overfitting. Random horizontal or vertical flipping, angle rotation, and scaling were performed on the original samples using Python scripts, and the training samples were expanded to at least 200 per category. The rotation, flipping, shearing, and scale transformations were conducted using some arbitrary bounded values that were uniformly distributed in a specific range for the generation of new images. For example, the shear and rotation ranges were set to 0.2 and ± 15°, respectively, where the scale range was increased from 0.9 to 1.1.

Then, using the approach introduced in Sect. 2.4, we conducted both model training and testing on the commodity label images, as shown in Fig. 7. To compare the algorithms, we selected six powerful, lightweight CNN models that include, MobileNet, MobileNet V2 [35], NASNetMobile, EfficientNet-B0, PeleeNet [36], and DenseNet as the baseline algorithms for a detailed comparative analysis. Considering the resource constraints of model training, these models developed using the method of transfer learning except for PeleeNet, which adopts the dense structure of DenseNet like MS-DenseNet. Therefore, they should have the same advantages as DenseNet and were selected together to train models from scratch for the comparison; besides, there is no official release version of TensorFlow for the PelleNet, and the unofficial pre-trained model performs poorly. For the transfer learning models,

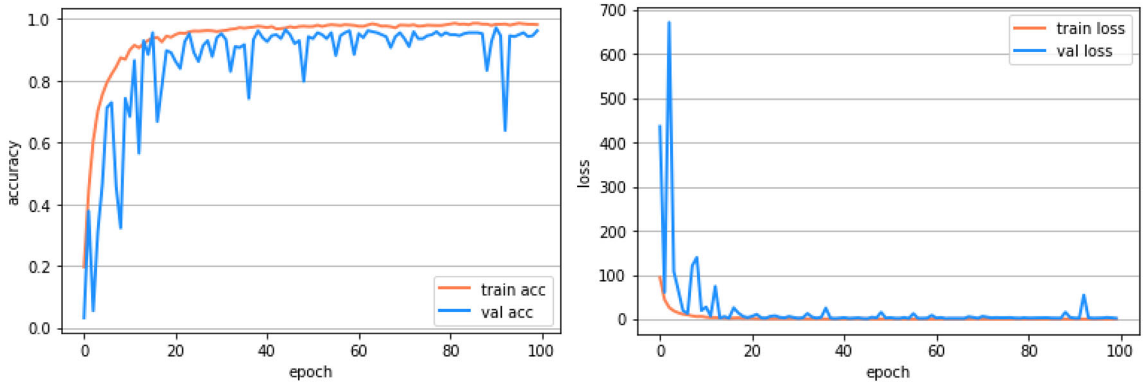


Fig. 7 The performance of model training for the proposed method

they were trained using pre-trained weights from ImageNet (https://keras.io/api/applications/), and the upper layers were replaced by a new fully-connected Softmax layer with the practical number of classes for the classification. Particularly, in order to ensure a fair comparison, the parameters of all the models are kept the same, such as the hyper-parameters of training epochs, minimum batch-size, learning rate, and the used optimizer, etc. In this manner, a number of experiments were performed by training multiple CNN models on the commodity label images. Each experiment runs for 30 and 100 epochs with a batch size of 64, a learning rate of 0.01, and a weight decay of 0.0001. Here, the well-known SGD (stochastic gradient descent) is utilized as an optimizer for updating weights.

Considering the statistics of accurate detections and misdetections, we measure model performances with metrics like *Accuracy*, *F1-Score, and Precision*, as defined by:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{19}$$

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20}$$

$$\text{F1-Score} = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}} \tag{21}$$

where true positive (TP), false negative (FN), false positive (FP), and true negative (TN) are the numbers of accurately predicted samples, inaccurately predicted samples, falsely detection, and the sum of accurately predicted samples in all categories except the relevant ones. Table 2 displays the results of various methods.

The proposed approach has achieved impressive performance in model training, as can be seen in Fig. 7. It performs well with the highest accuracy while having the lowest log-loss. Furthermore, the proposed approach has achieved results comparable to the other state-of-the-art approaches, as shown in Table 2. In addition, the test accuracies of the proposed MS-DenseNet hit 93.23% and 95.16% after training for 30 and 100 epochs, respectively. These are the highest accuracies among all the reported models except for the DenseNet-121, which comparatively has a large volume that requires more storage and computational resources. By contrast, the proposed approach outperforms the other lightweight neural network methods and achieves comparable accuracy with less overhead than deep CNNs. More than that, the Friedman statistical test [37, 38], which is among the most effective statistical significance tests for comparing different algorithms, was used to validate the performance of the proposed approach. Let r_i^j be the sorting of the *J*-th algorithm on the *I*-th metrics, then the average ranking is written in Eq. (22).

$$R_j = \frac{1}{N}\sum_i r_i^j \tag{22}$$

where R_j is the average ranking of the *j*-th algorithm, r_i^j is the ranking of the *j*-th algorithm on the *i*-th metrics. The null hypothesis of the Friedman test is that there is no difference in the performance of each algorithm. In other terms, their average ranking is the same. Thereby, the Friedman test can be calculated using Eq. (23).

$$F_F = \frac{(N-1)x_F^2}{N(k-1) - x_F^2} \tag{23}$$

where k represents the number of algorithms, N is the number of metrics, and x_F^2 is determined by Eq. (24) accordingly (R is the rank). The F_F follows the F distribution with the freedom degree of k—1 and (k-1)(N-1).

$$x_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{24}$$

Based on the abovementioned processes, the Friedman statistical test can be performed on the test results of different algorithms. It is easy to know that the k and N are equal to 7 and 3 here, and thereby the values of x_F^2 and F_F are calculated as 15.28 and 5.63, respectively. With the

Table 2 Model training results for the reported methods on the commodity label image dataset

Pre-trained models	30 epochs			100 epochs				
	Training accuracies %	Test accuracies %	Training losses	Training accuracies %	Test accuracies %	Training losses	Test losses	Avg_Rank
MobileNet	96.23	91.33 (5)	1.2674	98.52	91.87 (7)	1.0610	1.2207 (6)	6.00
MobileNet-V2	97.13	92.67 (3)	1.2691	98.86	94.67 (3)	1.0327	1.1331 (4)	3.33
NASNetMobile	94.17	89.38 (7)	1.4924	97.85	92.50 (6)	1.0588	1.2156 (5)	6.00
EfficientNet-B0	95.73	92.00 (4)	0.8566	96.65	92.67 (5)	0.5760	0.6766 (3)	4.00
PeleeNet	98.40	91.25 (6)	2.7749	99.03	93.75 (4)	1.6937	1.9008 (7)	5.67
DenseNet-121	97.85	94.37 (1)	0.4350	98.78	95.63 (1)	0.1516	0.2161 (2)	1.33
Proposed approach	97.13	93.23 (2)	0.0973	98.23	95.16 (2)	0.0377	0.1177 (1)	1.67

freedom degree of $k-1$ and $(k-1)(N-1)$, the F_F value is found as $F(k-1, (k-1)(N-1)) = F(6, 12) = 3.00$ from the critical value table of F distribution ($a = 0.05$). As a consequence, the null hypothesis is rejected since the 5.63 is greater than 3.00, and the differences between these algorithms are statistically significant. Thus, on the basis of the above analysis, the new unseen samples can be chosen to perform the class prediction of commodity label images using the proposed method. Table 3 displays the partial prediction results along with the indicator calculation, and the corresponding ROC curve, as well as the confusion matrix, are depicted in Fig. 8.

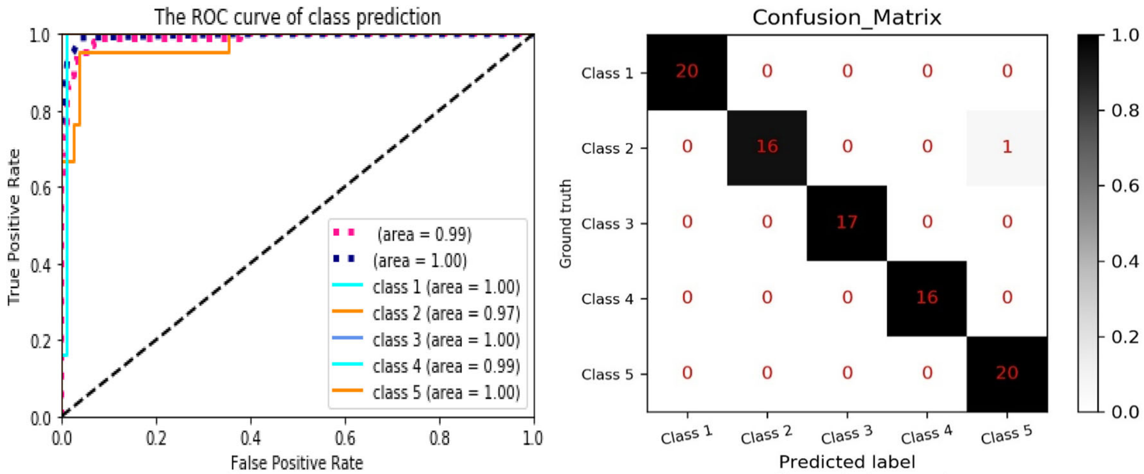
Figure 9 reveals the partially identified samples. The upper and lower images are the original and identified samples, respectively. As seen in Fig. 9a, b, d, the predicted labels are compatible with the actual ones, which the proposed approach identifies accurately, and the recognition probabilities are all above 0.90. It means that these samples are accurately identified with a higher probability. On the contrary, there are also individual sample images mistakenly classified by the proposed method, as given in Fig. 9c. The explanation for this is that the individual samples are mixed into different categories. The varied shades and uneven lighting strengths also impact the extraction of the features and can result in inaccurate classifications. Though some individual samples are misclassified, the proposed approach accurately recognizes the majority of the commodity label images. As illustrated in Fig. 8a, the ROC curve of each class that is close to the upper left corner denotes the ideal operating characteristics of the proposed method. The confusion matrix in Fig. 8b also illustrates that the proposed approach accurately identifies the majority of the samples belonging to each individual class. The predicted accuracy of each class is not less than 96.00%, and the average accuracy reaches 97.60%, as listed in Table 3. It demonstrates that the proposed MS-DenseNet is capable of recognizing the commodity label images.

3.2 Experiments on public datasets

We conducted several experiments on the open-source datasets to verify the validity of the proposed approach. These datasets include Scene15 [39], Caltech101 [40], and

Table 3 Evaluation indicators of identified results

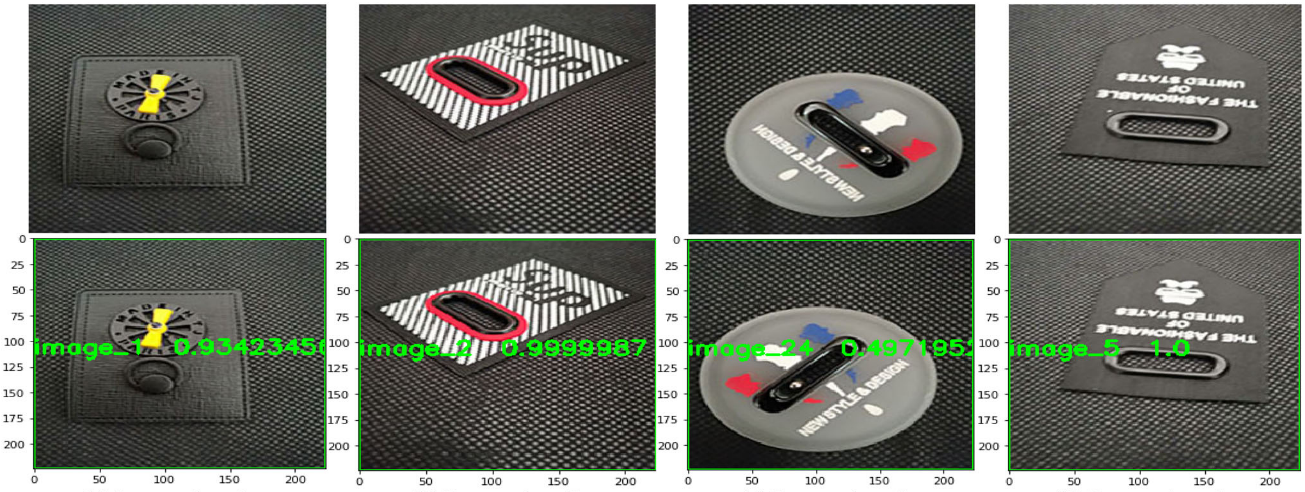
Types	Identified samples	Correct samples	Accuracy (%)	Precision (%)	F1-Score (%)
Class 1	20	20	100.00	100.00	100.00
Class 2	20	16	96.00	100.00	88.89
Class 3	20	17	97.00	100.00	91.89
Class 4	20	16	96.00	100.00	88.89
Class 5	20	20	99.00	95.23	97.56
Average	–	–	97.60	98.89	93.68



(a) ROC curve

(b) Confusion matrix

Fig. 8 ROC curve and confusion matrix of class prediction



(a) Image class 1

(b) Image class 2

(c) Image class 3

(d) Image class 5

Fig. 9 The predicted samples of commodity label images

Nus-wide [41]. Among them, the Scene15 dataset contains 4,485 images in total, which is determined by 15 scene categories including office areas, bedrooms, and forests; approximately 200–400 images are in each category for this dataset. Caltech101 dataset contains object images around with 101 categories, where most of the categories have at most 50 images while others have 40 to 800 images. Note that the dimension of each image is around 300 × 200 pixels. The Nus-wide dataset is a popular dataset containing around 269,648 images with a total of 5,018 unique tags, where each image is tagged from flicker. Similar to the experiments conducted in Sect. 3.1, we first perform the experiments on the Scene15 dataset and divide the dataset into the training and test sets with a proportion of 8:2. After 30 epoch training, the performance investigation for the proposed procedure compared with the results in the literature [39, 42–45] is displayed in Table 4. Note that the training samples are enriched to 1,000 images per category using the data augmentation scheme. The proposed method reaches a reasonably higher accuracy rate, which offers promising results when compared to other existing techniques. Experimental findings reveal that the proposed MS-DenseNet can effectively recognize these image types.

Furthermore, we compared the proposed approach with the baseline algorithms on the Caltech101 and Nus-wide datasets. The categories of the barrel, bonsai, camera, watch, and butterfly in the Caltech101 dataset were randomly selected to perform the test along with the airplane_flying, cat, face, flower, and waterfall categories in the Nus-wide dataset. To fit the models, all of the samples were evenly set to a fixed-dimension, and the categorical variable was one-hot encoded to be used in the models. Furthermore, a subset of the original images was reserved

Table 4 Comparative analysis of the proposed approach with other state-of-the-art

ID	Researches	Models	Classification accuracy (%)
1	Lazebnik et al. [39]	K-SPM	81.40
2	Yang et al. [42]	Sparsecoding-SPM	80.28
3	Rasiwasia et al. [43]	Low-dimensional Semantic Spaces	72.20
4	Li et al. [44]	ObjectBank	80.90
5	Sun et al. [45]	PCA-based CNN-SVM	81.06
6	Proposed approach	MS-DenseNet	83.13

for testing the model in order to see how the proposed technique works on unseen samples. In addition, the training and test sets were split in an 8:2 ratio, and the data augmentation scheme assured that at least 200 training samples were available for each category. For comparative study, the six baseline algorithms MobileNet, MobileNet V2, NASNetMobile, EfficientNet-B0, PeleeNet, and DenseNet were used. These multiple models were trained on public datasets for numerous experiments. Table 5 displays the results of model training. Figures 10, 11 display the raw images and the augmented sample images, respectively.

From Table 5, it can be seen that the proposed method has achieved a significant performance gain compared to other reported approaches, and after 30 and 100 epochs of training, the test accuracy of our approach reaches 72.29% and 84.71%, respectively. Except EfficientNet-B0 and DenseNet-121, which have a relatively large number of parameters and consume more memory, the proposed method achieves the top performance. Similarly, the Friedman test was applied to the relevant findings to determine the statistical significance of any differences, where the values of x_F^2 and F_F were separately calculated as 9.29 and 3.42. As described in Sect. 3.1, it is not difficult to find that the critical value of F distribution is equal to 3.00 ($F(k-1,\ (k-1)(N-1)) = F(6,\ 12) = 3$). Thus, the null hypothesis is rejected since 3.42 is greater than 3.00, and the Friedman statistical test demonstrates that the improved results are statistically significant. Overall, the main reason behind the solid performance of the proposed procedure is that the MS-DenseNet retains the structure and composition of the transition layer and replaces standard convolution with a depthwise separable convolution. Further, the SE blocks are incorporated in the networks, which improves the model efficiency and fully takes advantage of the channel relationship features. Additionally, in model training, a two-stage progressive training strategy is used to ensure that the network's weight parameters are properly optimized. Other CNNs, on the other hand, do not attain optimal performance even with pre-trained weights. As a result, the network training performed through the proposed approach results in an optimal model that accurately classifies the new unseen images. The confusion matrices of the observed results are depicted in Fig. 12, whereas the

Table 5 The results of model training on public dataset

Pre-trained models	30 epochs			100 epochs				
	Training accuracies %	Test accuracies %	Training losses	Training accuracies %	Test accuracies %	Training losses	Test losses	Avg_Rank
MobileNet	85.35	71.32 (4)	0.9433	95.51	83.75 (5)	0.5429	0.9557 (3)	4.00
MobileNet-V2	87.11	72.68 (2)	0.9324	96.68	83.92 (4)	0.5375	1.0169 (4)	3.33
NASNetMobile	80.27	67.88 (7)	1.2748	91.99	81.95 (7)	0.8443	1.1046 (5)	6.33
EfficientNet-B0	78.32	73.68 (1)	1.3625	86.91	86.83 (2)	1.1708	1.3390 (6)	3.00
PeleeNet	98.20	70.38 (5)	3.9117	100.00	83.44 (6)	2.7837	1.9984 (7)	6.00
DenseNet-121	84.21	70.07 (6)	0.9154	96.12	87.99 (1)	0.3668	0.6017 (1)	2.67
Proposed approach	92.20	72.29 (3)	2.9624	98.60	84.71 (3)	0.4862	0.8675 (2)	2.67

(a) The image samples in Caltech-101dataset.

(b) The image samples in the Nus-wide dataset.

Fig. 10 The original image samples

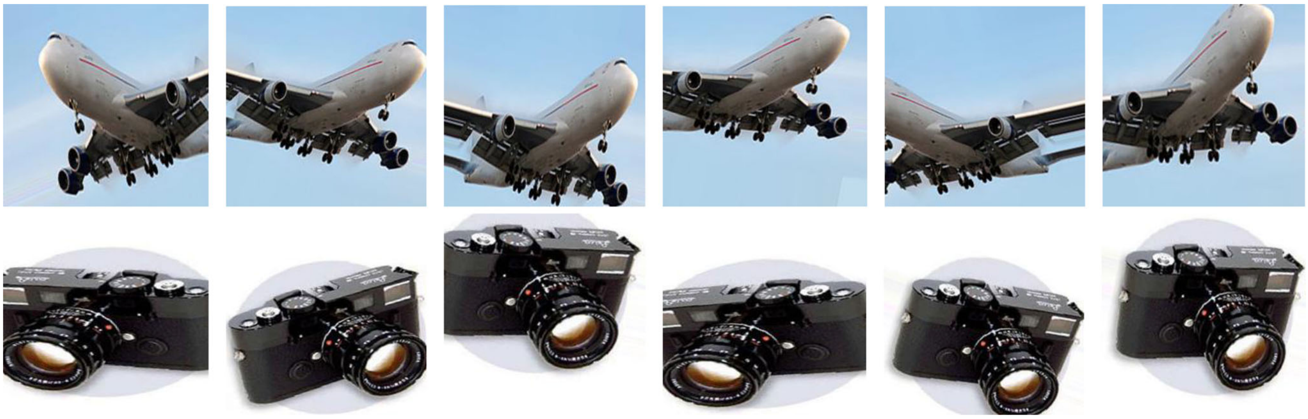


Fig. 11 Examples of augmented images

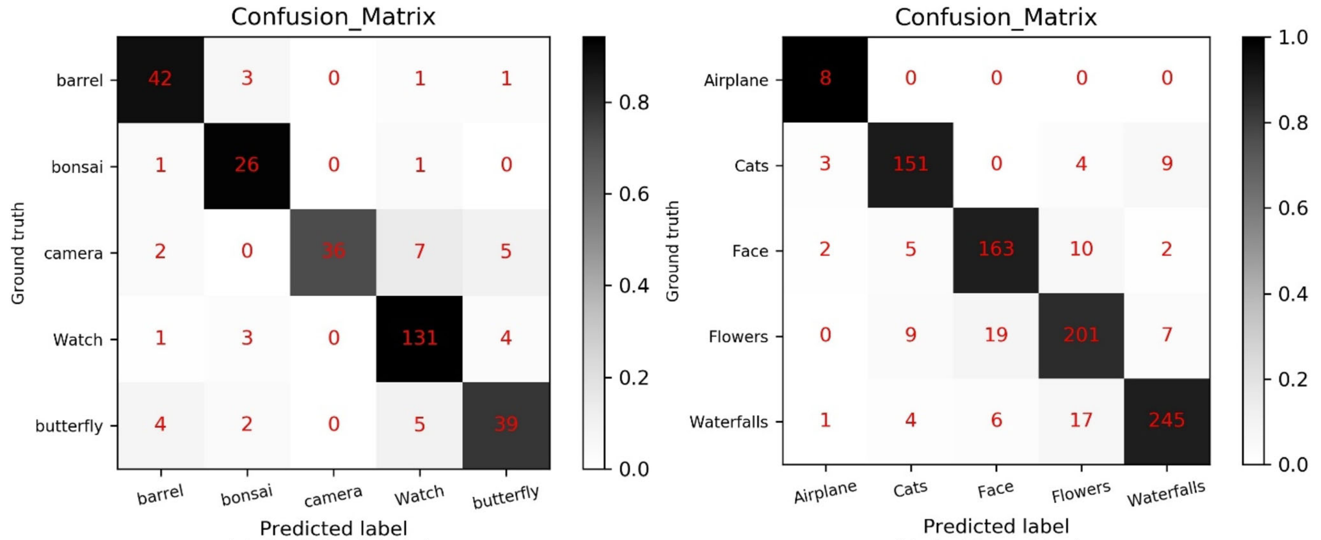
corresponding indicator measurements are listed in Tables 6, 7.

From Fig. 12, it can be observed that a few samples are misclassified; however, most of them are accurately identified. The average recognition *Accuracy* achieves 94.90% on the Caltech101 dataset while 95.45% on the Nus-wide dataset, as shown in Tables 6, 7. Also, both the precision and *F1-Score* reach no less than 86.00% on the test set, which demonstrates that the proposed architecture has successfully performed the image classification on the public dataset. Based on the experimental analysis, it can be assumed that the proposed approach is useful to commodity label identification and even applicable to other relevant fields, including online target identification, quality control, and fault detection, etc.

4 Conclusions

Corporations are facing various challenges in the digital economy, and digital transformation is a crucial process that potentially impacts the whole corporation. It supports in considering the utility of data to maintain and grow businesses in the fast-growing nature of the modern digital world, and corporations go through digital transformations to stay relevant in the digital economy era. The accurate identification of commodity labels through digital images is rather needed to support retail business automation and digital transformation. Deep learning techniques, particularly CNNs, have achieved much progress in addressing many technological issues involved with image recognition.

Therefore, based on popular mobile neural networks, we propose a novel lightweight architecture aliased as MS-



(a) On Caltech101 dataset (b) On Nus-wide dataset

Fig. 12 The confusion matrices of identified results

Table 6 The identified results on the Caltech101 dataset

Object	Identified samples	Correct samples	Accuracy (%)	Precision (%)	F1-score (%)
Barrel	47	42	95.85	84.00	86.59
Bonsai	28	26	96.81	76.47	83.87
Camera	50	36	95.54	100.00	83.72
Watch	139	131	92.99	90.34	92.55
Butterfly	50	39	93.31	78.00	78.79
Average	–	–	94.90	86.98	87.26

Table 7 The identified results on the Nus-wide dataset

Object	Identified samples	Correct samples	Accuracy (%)	Precision (%)	F1-score (%)
Airplane	8	8	99.30	57.14	72.72
Cats	167	151	96.07	89.34	89.88
Face	182	163	94.80	86.24	87.87
Flowers	236	201	92.38	86.63	85.89
Waterfalls	273	245	94.68	93.15	91.41
Average	–	–	95.45	88.58	88.63

DenseNet, which has a limited model size and comparable precision to recognize commodity label images. The current advanced DenseNet was chosen as the backbone network, and the depth-separable convolution was substituted for conventional convolution in dense blocks to compress the model size and allow better use of model parameters. Then, the SE blocks were incorporated in the proposed network to highlight the useful feature channels while suppressing the unwanted feature channels, which made good use of interdependencies between channels and realized the maximum reuse of inter-channel relation, improving model performance.

The experimental results present significant performance gains on both open and our local datasets. It demonstrates the superiority of the proposed approach over other state-of-the-art techniques for the identification of commodity label images and the image classification in other fields. Due to the cost of configuring hardware, we do not build the pre-trained model on ImageNet for transfer learning. In the future development, we consider adding more hardware devices for the pre-trained model. Particularly, we plan to implement the system on portable or mobile devices to automatically recognize the different available commodity details. Moreover, we would like to extend it to other real-world applications.

Acknowledgments The research is funded by the National Natural Science Foundation of China (61672439) and the Fundamental Research Funds for the Central Universities (20720181004). The authors also thank editors and all unknown reviewers for constructive suggestions.

Declarations

Conflicts of interest The authors declare no conflicts of interest.

References

1. Wang Y, Wang Z (2019) A survey of recent work on fine-grained image classification techniques. J Vis Commun Image Represent 59:210–214
2. Gomes SL et al (2017) Embedded real-time speed limit sign recognition using image processing and machine learning techniques. Neural Comput Appl 28(1):573–584
3. Li CH et al (2013) Algorithm research of two-dimensional size measurement on parts based on machine vision. Adv Mater Res 694:1945–1948
4. Gökmen V, Sügüt I (2007) A non-contact computer vision based analysis of color in foods. Int J Food Eng 3:1–13
5. Ciocca G, Napoletano P, Schettini R (2018) CNN-based features for retrieval and classification of food images. Comput Vis Image Underst 176:70–77
6. Geetharamani G, Pandian A (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput Electric Eng 76:323–338
7. Priyadharshini RA et al (2019) Maize leaf disease classification using deep convolutional neural networks. Neural Comput Appl 31(12):8887–8895
8. Miki Y et al (2017) Classification of teeth in cone-beam CT using deep convolutional neural network. Comput Biol Med 80:24–29
9. Mahbod A et al (2020) (2020) Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification. Comput Methods Programs Biomed 193:105475
10. Mondal S, Bours P (2017) A study on continuous authentication using a combination of keystroke and mouse biometrics. Neurocomputing 230:1–22
11. Duan Y et al (2017) SAR Image segmentation based on convolutional-wavelet neural network and markov random field. Pattern Recogn 64:255–267
12. Precup R-E et al (2020) Evolving fuzzy models for prosthetic hand myoelectric-based control. IEEE Trans Instrum Meas 69:4625–4636
13. Li X et al (2020) Fault diagnostics between different type of components: a transfer learning approach. Appl Soft Comput 86:105950
14. Ma J et al. (2019) Machine learning based cross-border E-commerce commodity customs product name recognition algorithm. In: Pacific Rim International Conference on Artificial Intelligence. Springer, Cham, pp 247–256
15. Ahmed MU et al (2019) A machine learning approach to classify pedestrians' events based on IMU and GPS. Int J Artif Intell 17(2):154–167
16. Zhang T, Chen E (2019) Product recognition algorithm based on HOG and bag of words model. In: 2019 8th International Symposium on Next Generation Electronics (ISNE), pp 1–3. IEEE
17. Kussul N et al (2017) Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci Remote Sens Lett 14(5):778–782
18. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
19. Ji Y et al (2019) Graph model-based salient object detection using objectness and multiple saliency cues. Neurocomputing 323:188–202
20. Zou X et al (2020) Multi-task cascade deep convolutional neural networks for large-scale commodity recognition. Neural Comput Appl 32(10):5633–5647
21. Chen C, Yang R, Wang C (2017) Research and realization of commodity image retrieval system based on deep learning. In: International Symposium on Parallel Architecture, Algorithm and Programming, vol 729, pp. 376–385. Springer, Singapore
22. Cao Z, Shaomin Mu, Dong M (2020) Two-attribute e-commerce image classification based on a convolutional neural network. Vis Comput 36:1619–1634
23. Huang G et al. (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2261–2269
24. Howard AG et al. (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861. pp 1–9
25. Sifre L, Mallat S (2014) Rigid-motion scattering for image classification. Ph. D. thesis
26. Kaiser L, Gomez AN, Chollet F (2017) Depthwise separable convolutions for neural machine translation, arXiv preprint arXiv:1706.03059. pp 1–10.
27. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
28. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 315–323
29. Zoph B et al. (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8697–8710
30. Zhang X et al. (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6848–6856
31. Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks, arXiv preprint arXiv:1905.11946, pp 6105–6114
32. Lin T-Y et al. (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
33. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980, pp 1–15
34. Ghazi MM, Yanikoglu B, Aptoula E (2017) Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235:228–235
35. Sandler M et al. (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4510–4520
36. Wang RJ, Li X, Ling CX (2018) Pelee: a real-time object detection system on mobile devices. Advances in Neural Information Processing Systems, pp 1–10
37. Pereira DG, Afonso A, Medeiros FM (2015) Overview of Friedman's test and post-hoc analysis. Commun Statis Simul Comput 44(10):2636–2653
38. Irigaray D et al. (2019) Accelerating the calculation of Friedman test tables on many-core processors. In: Latin American High Performance Computing Conference. Springer, Cham, pp 122–135
39. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene

categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol 2. IEEE, pp 2169–2178

40. Li F-F, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: 2004 conference on computer vision and pattern recognition workshop. IEEE, pp 178–178
41. Chua T-S et al. (2009) NUS-WIDE: a real-world web image database from National University of Singapore. In: Proceedings of the ACM international conference on image and video retrieval, pp 1–9
42. Yang J et al. (2009) Linear spatial pyramid matching using sparse coding for image classification. In: 2009 IEEE Conference on computer vision and pattern recognition. IEEE, pp 1794–1801
43. Rasiwasia N, Vasconcelos N (2008) Scene classification with low-dimensional semantic spaces and weak supervision. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 1–6
44. Li L et al (2010) Object bank: a high-level image representation for scene classification & semantic feature sparsification. Adv Neural Inf Process Syst 23:1378–1386
45. Sun Y et al (2019) Image classification base on PCA of multi-view deep representation. J Vis Commun Image Repres 62(2019):253–258