



# Printer source identification of quick response codes using residual attention network and smartphones

Zhongyuan Guo<sup>a</sup>, Shiyuan Wang<sup>a,\*</sup>, Zhaohui Zheng<sup>b</sup>, Ke Sun<sup>c</sup>

<sup>a</sup> College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

<sup>b</sup> School of Mathematics and Physics, Wuhan Institute of Technology, Wuhan 430079, China

<sup>c</sup> Wuhan Meter Intelligent Technology Co., Ltd, Wuhan, Hubei 430014, China

## ARTICLE INFO

### Keywords:

Printer source  
Quick response code  
Bottleneck residual block  
Squeeze-excitation module  
Smartphones

## ABSTRACT

Quick response codes are widely utilized in commodity anti-counterfeiting, and their traceability via smartphones is an effective authentication method. However, quick response codes are susceptible to illegal duplication, and the images captured by smartphones are easily affected by changes in light and environmental noise, leading to unsatisfactory verification results. To address these issues, this paper introduces a novel approach by combining the squeeze-excitation attention module with bottleneck residual block. It presents a squeeze-excitation bottleneck residual network for printer source identification of quick response codes. The squeeze-excitation attention module pays more attention to the features that represent the printer attributes, reduces the interference of useless information, and has low computational consumption; while the bottleneck residual block has the advantages of few parameters, strong feature extraction capability, and good expandability. Thus, the performance can be improved effectively only with a small increasing in parameters. The experimental results verify that the proposed method achieves an accuracy of 98.77% under smartphone capture conditions, it outperforms other convolutional neural network-based methods in terms of identification accuracy. The deep learning model proposed in this paper can be generalized and applied to the printer source identification of paper content in the civil, criminal investigation and judicial fields.

## 1. Introduction

With the rapid development of 5G information technology and the rapid popularization of smartphones, quick response (QR) code, as shown in Fig. 1, has been widely used in many fields. Generally, QR code is printed on the packaging of various products such as printed matter, food, medicine, cosmetics, tobacco and alcohol, tea, and daily necessities, acting as anti-counterfeiting labels, so that consumers can easily scan these QR codes through smartphones, and identify the authenticity of products while obtaining products information (Focardi et al., 2019; Aini et al., 2020). However, QR code itself does not have anti-counterfeiting function, and is easy to be forged by illegal copying (Chen et al., 2019). Forged QR code is bound to counterfeit and shoddy product, then circulated into the market, which is likely to cause economic disputes and criminal cases. Therefore, it is an urgent task to develop an effective and appropriate testing tool to identify the printer source of QR code.

When a digital image of QR code is printed, the QR code takes on the

inherent characteristics of the printer, also known as the printer's "fingerprint" (Ali et al., 2003). Note that each printer's fingerprint is unique (Jain et al., 2020; Bibi et al., 2019; Oliver and Chen, 2002), which provides a basis for the identification of printer source of QR code. The rapid development of smartphones has made it relatively easy to obtain digital images of QR codes, and identifying the printer source of QR codes through smart phones is a fast, convenient and reliable method. However, compared with the previous identification method of obtaining digital images by scanner, the way of shooting and collecting with smartphones is affected by factors such as light changes and ambient noise, and the quality of the captured digital QR code images is degraded to a certain extent, which makes the effect of printer source identification unsatisfactory, and there is an urgent need to design an effective solution.

In order to meet the public's demand for convenient anti-counterfeiting authentication using smartphones, this paper proposes a network that combines bottleneck residual blocks (BRB) (He et al., 2016) and Squeeze-Excitation (SE) attention mechanism (Hu et al.,

\* Corresponding author.

E-mail address: [wsy@swu.edu.cn](mailto:wsy@swu.edu.cn) (S. Wang).



Fig. 1. QR code.

2018) to deal with the printer source identification of QR codes captured by smartphones. The use of SE attention mechanism allows the proposed network to pay more attention to the information related to the printer attributes and suppress the influence of useless information, which enables the proposed convolutional neural network to outperform the state-of-the-art method in terms of accuracy, with an identification accuracy of 98.77%. The main contributes of this paper are as follows:

- (1) In order to fully extract the “fingerprint” features that represent printer attributes while suppressing the interference of irrelevant information such as noise, this paper combines the Bottleneck Residual Block and the Squeeze-Excitation module to design a Squeeze-Excitation Bottleneck Residual Block (SE-BRB).
- (2) The SE-BRB is used as the basic unit to propose a QR code printer identification network under the conditions of smartphone shooting and acquisition.
- (3) Nine printers and five smartphones are used to build a self-built QR Code data set, which makes up for the fact that no publicly available smartphone-captured data set for identifying the printer source of QR code exists yet.
- (4) We evaluate the performance of the proposed method against the existing methods on our self-constructed data set, the results show that the proposed method provides better accuracy in printer source identification.

The paper is organized as follows. Section 2 is related works. Section 3 is the proposed method, including the composition, principle and overall identification process of the proposed squeeze-excitation bottleneck residual network. Section 4 is the experiment results and discussion, including experimental settings, data set production, ablation experiment and comparative experiments. The last section contains conclusion and future improvements.

## 2. Related works

Most of the existing printer source identification methods are designed for documents, calligraphy and paintings, and banknotes. For example, Tsai (Tsai et al., 2016) used gray level co-occurrence matrix (GLCM) and discrete wavelet transform method to extract the features of scanned Chinese characters to find the print source. Choi (Choi et al., 2013) proposed a color laser printer source identification method based on noise texture analysis and support vector machine (SVM) classifier in order to detect the source of color printed matters such as artworks and banknotes. Zhou (Zhou et al., 2016) studied printer source identification independent of text content, and proposed a method based on synthetic texture analysis, using fast fourier transform (FFT) and GLCM method for feature extraction, and then using SVM for classification. Mikkilineni A K (Mikkilineni et al., 2010) developed a printer source identification system using the intrinsic signature of the printer combined with texture

and band features. Nguyen (Nguyen et al., 2019) performed a statistical analysis of printed patterns at microscopic scale, used shape descriptors as features, and developed a classification algorithm based on SVM and random forests for printer source identification. Hamzehyan (Hamzehyan et al., 2021) proposed a new method of printer source identification based on the Gaussian Mixture Model-Universal Background Model (GMM-UBM) method and joint factor analysis by modeling the main local binary pattern in the total variable printer space and extracting secondary features based on joint factor analysis. However, the above methods are manually designed feature extraction and feature selection methods based on expert experience, which is time-consuming. In addition, random selection of training and testing samples leads to complex processing process, and it is necessary to repeat the whole process several times, which will undoubtedly consume a lot of manpower, material and financial resources.

In recent years, deep learning (LeCun et al., 2015; Theodoridis, 2015; Jiang et al., 2022) has been widely used in computer vision, image processing and other fields. Convolutional neural network (CNN) (Gu et al., 2018; Khan et al., 2020) is one of the most representative algorithms in the field of deep learning, and has made breakthroughs in computer vision issues, e.g., image classification (Chan et al., 2015; He et al., 2019), object detection (Druzhkov and Kustikova, 2016; Zhao et al., 2019; Roy and Bhaduri, 2023), and image segmentation (Minaee et al., 2021; Wang et al., 2018; Ghosh et al., 2019). Some preliminary researches on the application of CNN to the identification of printer sources of documents or photos have been performed. Anselmo Ferreira (Ferreira et al., 2017) designed a deep learning method for the identification of laser printer attributes based on the analysis of small block character patterns, using multiple shallow CNNs to learn multiple representations of the same character and fusing the features. A special equipment was used for data collection, which will undoubtedly consume a lot of financial, material and time resources. Do-Guk Kim (Kim et al., 2019) proposed a color laser printer identification method based on deep neural network and cascaded learning, used adversarial training to train the refiner network, decomposed the halftone color of the synthesized pattern data set, trained the decomposition network through the refined data set, and transferred the trained knowledge to the printer identification network to ensure high-precision identification results. However, it is designed for source color printer, and its halftone decomposition method is not applicable to the black-and-white image of anti-counterfeiting QR code. Min-Jen Tsai (Tsai et al., 2019) designed a shallow CNN to perform printer source authentication on images and document data collected by microscopes and scanners, and achieved roughly the same effect as shallow machine learning. However, its recognition performance will be significantly degenerated under the condition of mobile device acquisition. Guo (Guo et al., 2020) proposed a convolutional neural network for printer source identification of QR codes scanned by scanners, named PSINet, which is mainly composed of residual modules and achieved high identification accuracy on eight printers. This is because the scanner is specialised equipment, the QR code image scanned by scanner is clear and of high quality, so it is easy to extract the features that represent the attributes of the printers. However, the scanner is large in size, inconvenient to carry, and has a limited application range.

In summary, most of the current studies on printer source identification use specialised equipment such as scanner or microscope, which has the problems of large size and inconvenience of carrying, and there are few studies on smartphone-oriented QR code printer source identification. With the popularity of smartphones and the convenience of QR code image acquisition, printer source identification of anti-counterfeiting QR code captured by smartphones has become an urgent need for commodity anti-counterfeiting and document forensics, and has important research value and application significance. Therefore, we design a new deep learning solution for QR code images captured by smartphones, aiming to achieve universal and convenient QR code authentication.

### 3. The proposed identification method

#### 3.1. Bottleneck residual block

The BRB was proposed by He Kaiming (He et al., 2016) in 2015, and ResNet built on BRB won the championship in the ILSVRC competition that year. The BRB module is formed by stacking  $1 \times 1$ ,  $3 \times 3$  and  $1 \times 1$  convolutional layers, as shown in Fig. 2, 256 channels are reduced to 64 channels by the first  $1 \times 1$  bottleneck layer, and then restored to 256 channels by the last  $1 \times 1$  bottleneck layer; the *ReLU* activation function (Agarap, 2018) enhances the nonlinearity of the module and alleviate the degradation problem due to deeper network layers and degradation of the weight matrix. The convolution and activation operations are denoted by  $F(x_l, W_l)$ , and the BRB module learns the residual term during the training process, which can be expressed as Eq. (1):

$$F(x_l, W_l) = H(x_{l+1}) - x_l \quad (1)$$

When  $F(x_l, W_l) = 0$ , there is  $H(x_{l+1}) = x_l$ , which is the identity mapping. During iterative training, the output of the  $L$ -th layer can be expressed as follows:

$$x_L = \sum_{i=1}^{L-1} F(x_i, w_i) + x_1 \quad (2)$$

From Eq. (2), it can be seen that any layer in the residual network can be represented by the output of the previous layer  $x_l$  and the residual part  $\sum_{i=1}^{L-1} F(x_i, w_i)$ .

Meanwhile, the design of identity mapping branch transmits the shallow extracted features directly backward, which achieves feature reuse without increasing the parameters and avoids the loss of features in the CNN. According to the chain derivative rule, the back propagation of the residual network can be expressed as follows:

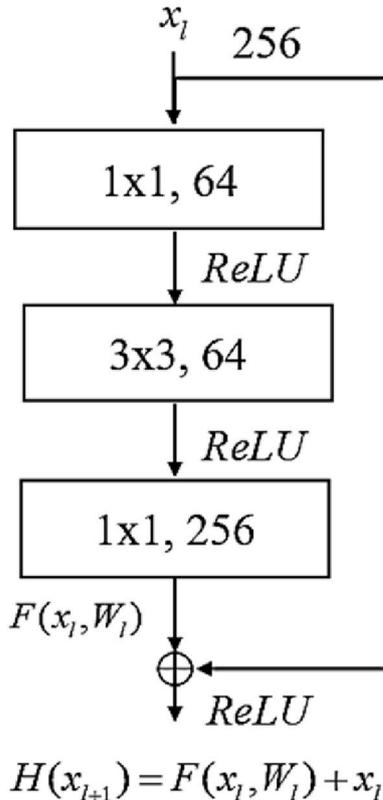


Fig. 2. Structure of bottleneck residual block.

$$\frac{\partial J}{\partial x_l} = \frac{\partial J}{\partial x_L} \frac{\partial x_L}{\partial x_l} = \frac{\partial J}{\partial x_L} \left( 1 + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} F(x_i, w_i) \right) = \frac{\partial J}{\partial x_L} + \frac{\partial J}{\partial x_L} \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} F(x_i, w_i) \quad (3)$$

Where  $L$  represents the number of deep layers;  $l$  represents the number of shallow layers;  $x_i$  and  $w_i$  represent the input and weight of the  $i$ -th layer, respectively;  $x_L$  represents the output of  $L$ -th layer;  $J$  represents the loss function;  $\partial J / \partial x_L$  indicates that the gradient of the  $L$ -th layer is directly transferred to the shallow  $i$ -th layer. The value of the component  $\frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} F(x_i, w_i)$  determines whether the problem of gradient disappearance will occur during the backpropagation of the residual network. Due to the non-negativity of the activation function *ReLU*, the chain derivation result in the brackets of formula (3) is non-negative. Hence, the value of the part in the brackets is greater than 1, thus avoiding the gradient disappearance and solving the degradation problem of CNN with no increasing of the parameters and computation of the CNN at the same time.

#### 3.2. Squeeze-excitation module

SE module is shown in Fig. 3, which is divided into two parts: Squeeze and Excitation. The Squeeze part compress the features of dimension  $H \times W \times C$  using global mean pooling to obtain dimension  $1 \times 1 \times C$ , where  $H$  denotes height,  $W$  denotes width, and  $C$  denotes the number of channels.  $H \times W$  becomes one-dimensional after squeeze operation, which makes the global features of  $H \times W$  assigned to the squeezed parameters and increases the perceptual area. The Excitation part is a fully connected layer that takes the squeeze part  $1 \times 1 \times C$  as input and predicts the importance of each channel. Then, the *Scale* operation greatly enhances the main features and enables the effective improvement of the recognition accuracy of the CNN model.

The principle of the SE module is as follows. First, the input features  $X$  are mapped to  $U$  by the given mapping condition  $F_{tr}$  with  $X \in \mathbb{R}^{H \times W \times C}$  and  $U \in \mathbb{R}^{H \times W \times C}$ , which can be described by

$$F_{tr} : X \rightarrow U, X \in \mathbb{R}^{H \times W \times C}, U \in \mathbb{R}^{H \times W \times C} \quad (4)$$

Specifically,  $F_{tr}$  is a standard convolution operation, which is calculated as follows:

$$u_c = v_c * X = \sum_{s=1}^C v_c^s * x^s \quad (5)$$

where  $*$  denotes the convolution operation,  $v_c = [v_c^1, v_c^2, \dots, v_c^C]$ ,  $X = [x^1, x^2, \dots, x^C]$ .  $v_c^s$  is a two-dimensional spatial kernel representing a single channel of  $v_c$ , which acts on the corresponding channel of  $X$ . The output  $u_c$  is generated by summing all channels, with channel dependencies implicitly embedded in  $v_c$  and these dependencies entangled with the spatial correlations captured by the filter.

Then, after the  $F_{tr}$  operation, we get  $U$ , followed by the Squeeze operation, which is essentially global average pooling:

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

The  $F_{sq}$  operation converts the input  $H \times W \times C$  to the output  $1 \times 1 \times C$ , turning each two-dimensional feature channel into a real number that somehow has a global field of perception and the dimensionality of the output matches the number of feature channels of the input.

Next the Excitation operation is geared towards recalibrating the channel information after compression of the feature information resulting from the compression operation, as shown by

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

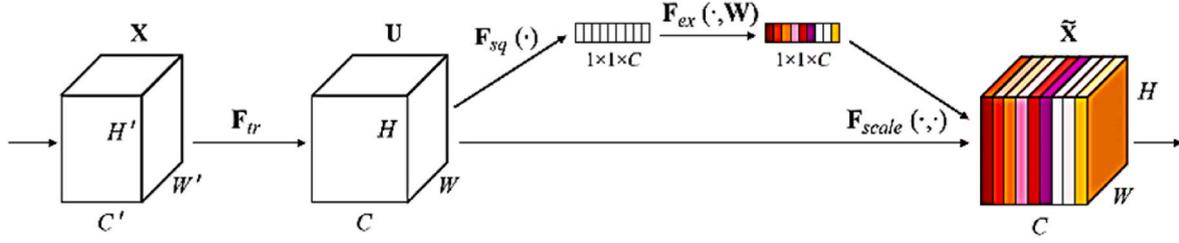


Fig. 3. Squeeze and excitation module.

where  $z$  denotes the result after the Squeeze operation,  $W_1 \in R^{c \times c}$ ,  $W_2 \in R^{c \times \frac{c}{r}}$ , with  $r$  being the scaling parameter which is set to 16 in this paper and its purpose is to reduce the number of channels and reduce the amount of calculation.  $W_1 z$  indicates the operation of the fully connected layer.  $\delta$  represents the activation function *Relu*, and the output dimension after processing in this step remains unchanged, and then multiplied by  $W_2$ , which also represents the operation of the fully connected layer with the output dimension being  $1 \times 1 \times C$ .  $\sigma$  represents the activation function *sigmoid*, and the output after processing is  $s$ .

The final output of the SE module is obtained by rescaling the transformation output  $U$  with the activations, i.e.,

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

where  $u_c$  is a two-dimensional matrix,  $s_c$  represents the weights, and the  $F_{scale}$  operation is equivalent to multiplying each value in the matrix  $u_c$  by  $s_c$ .

### 3.3. SE-BRB module

By adding the Squeeze and Excitation module after the last  $1 \times 1$  convolutional layer of the BRB module, the SE-BRB module can be formed, as shown in Fig. 4, which can explicitly represent the relationship between the feature channels, so that the information between the channels can be interacted. The rescaling method are used for the extracted features, and then the weight are obtained according to the importance of the channels, that is, automatically obtain the importance of the feature channel through CNN's learning. The complexity of the CNN is reduced significantly, and the recognition accuracy of the CNN is therefore improved.

### 3.4. Printer source identification network based on SE-BRB module

For the printer source identification of QR codes, since the differences between categories are relatively subtle, the input image data need to be divided into blocks to avoid the interference of content, and the size of the image block is small.

This paper proposes a printer source identification network based on SE-BRB module, named SE-BRB-Net, with a total of 14 layers, as shown in Fig. 5. The SE-BRB-Net mainly contains a input layer, a  $3 \times 3$  convolutional layer, an average pooling layer with a kernel size of  $3 \times 3$ , four SE-BRB modules, a global average pooling layer, and a fully connected layer. Each SE-BRB module is composed of a BRB and a SE module, as shown in the rectangular box in Fig. 5. The composition and principles of the BRB and SE module have been introduced in detail in Section 3.1 and Section 3.2 respectively, and the SE-BRB module is introduced in detail in Section 3.3. A Batch Normalization layer is introduced after each convolutional layer in the BRB module with the aim of accelerating the training process and improving the generalization ability of the model, the use of ReLU activation function increases the nonlinearity of the network. Connected after the fourth SE-BRB module is the global average pooling layer, which can reduce network parameters and speed up the training process.

Table 1 shows the parameters of the SE-BRB-Net, and the model

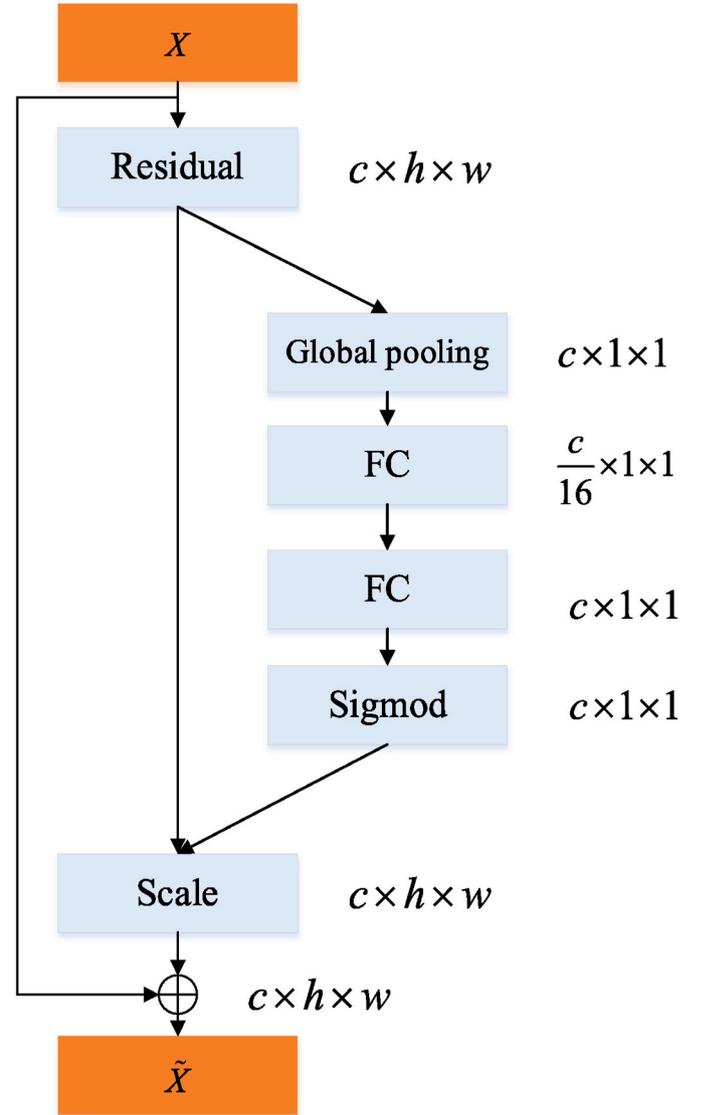


Fig. 4. SE-BRB module.

complexity is relatively low since both  $3 \times 3$  and  $1 \times 1$  sized convolutions are used, which is more suitable for solving the small-size image identification problem in the field of printer source identification. The Softmax function maps the feature vectors into probability distributions as shown in Equation (9). Since the number of used printers is 9, the output of the fully connected layer is set to 9.

$$f(z_i) = - \left( \frac{e^{z_i}}{\sum_{i=1}^n e^{z_i}} \right) \quad (9)$$

where  $i$  denotes the category label and  $n$  denotes the number of cate-

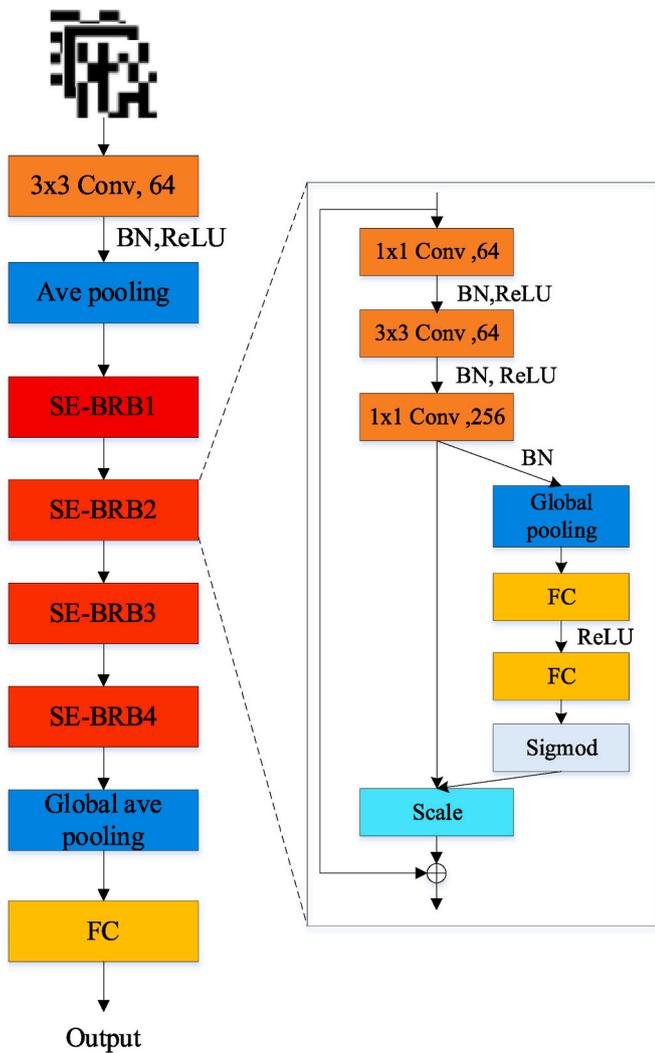


Fig. 5. The overall framework of SE-BRB-Net.

Table 1  
Parameters of SE-BRB-Net.

| Module             | The output feature map size | Parameters   |
|--------------------|-----------------------------|--|
| Input              | $64 \times 64 \times 1$     |  |
| Conv1              | $64 \times 64$              | $3 \times 3 \times 64$<br>Stride:1 Padding:1   |
| Ave_Pooling        | $32 \times 32$              | $3 \times 3 \times 64$<br>Stride:2 Padding: 0  |
| SE-BRB1            | $32 \times 32$              | $1 \times 1 \times 64$ Strdie:1<br>$3 \times 3 \times 64$ Strdie:1                                     |
| SE-BRB2            | $32 \times 32$              | $1 \times 1 \times 256$ Strdie:1<br>$1 \times 1 \times 64$ Strdie:1<br>$3 \times 3 \times 64$ Strdie:1 |
| SE-BRB3            | $32 \times 32$              | $1 \times 1 \times 256$ Strdie:1<br>$1 \times 1 \times 64$ Strdie:1<br>$3 \times 3 \times 64$ Strdie:1 |
| SE-BRB4            | $32 \times 32$              | $1 \times 1 \times 256$ Strdie:1<br>$1 \times 1 \times 64$ Strdie:1<br>$3 \times 3 \times 64$ Strdie:1 |
| Global_Ave_Pooling | $1 \times 1$                | -  |
| Softmax            | $1 \times n$                | -  |

gories.

### 3.5. The overall identification process

The overall flow of the proposed scheme is shown in Fig. 6, which mainly included two part: training process and testing process, which are shown in detail as follows:

- (1) Printing and collection of QR codes: The authentic QR code labels were obtained by printing digital QR codes using officially authorized printers. Forgery was performed by scanning and then printing on the basis of the authentic QR codes, and a total of eight all-in-one printers and copiers were used for forgery in this paper. In terms of QR code collection, five smartphones were used.
- (2) Image block preprocessing: The size of genuine QR codes and eight types of counterfeit QR codes are unified to  $512 \times 512$ , and are divided into training set, validation set, and test set according to the ratio of 3:1:1. The QR codes have been divided into blocks, and the size of the image block is determined to  $64 \times 64$  in this paper.
- (3) SE-BRB-Net training: Train the SE-BRB-Net, and obtain the printer source identification model through hyper-parameter optimization.
- (4) Model testing: The QR code image blocks in the test set are input to the CNN model for prediction, the prediction accuracy of each block is obtained, and then average the prediction accuracy rates of all image blocks to determine the printer source of the QR code to be tested.

## 4. Experimental results and discussion

### 4.1. Experimental setup

The hardware and software used for the experiments are shown in Table 2.

### 4.2. Data set production

At present, there is no publicly available data set for research on printer source identification of QR code collected from smartphones. Therefore, we built our own QR code data set. First, the officially authorized printer Toshiba e-studio 2051c- 11606695 was used to print 48 QR codes with different contents as a batch of genuine QR codes. Next, 8 printers of different brands and models were used as counterfeit printers, and each printer printed 48 QR codes with different contents. The brands and models of all printers are shown in Table 3.

Then, 5 mobile phones of different brands or models were used to

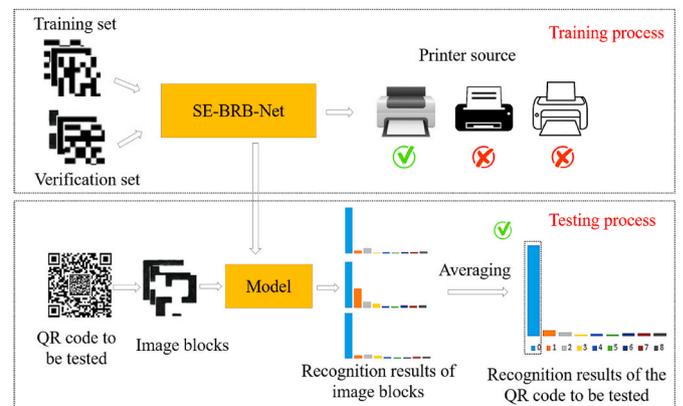


Fig. 6. Overall identification process.

**Table 2**  
Experimental platforms and parameters.

| Platforms               | Parameters                             |
|-------------------------|--|
| Notebook                | HP OMEN 17-w119TX                      |
| CPU                     | Intel Core i7- 7700HQ (2.80 GHz,16 GB) |
| GPU                     | Nvidia Geforce GTX1070 (8 GB)          |
| Operation System        | Windows10                              |
| Deep learning framework | Caffe and Digits 6.0                   |
| CUDA                    | 9.0                                    |
| CUDNN                   | 7.0.5                                  |

**Table 3**  
Brands and models of printers and number of QR codes.

| No.          | Brand   | Model                     | Number of QR codes |
|--------------|---------|---------------------------|--------------------|
| P0           | Canon   | image Runner ADVANCE 6575 | 48                 |
| P1           | Epson   | L15168                    | 48                 |
| P2           | Leovo   | MD7600                    | 48                 |
| P3           | RICOH   | Aficio MP7502-I           | 48                 |
| P4           | RICOH   | Aficio MP7502-II          | 48                 |
| P5           | RICOH   | Aficio MP8001             | 48                 |
| P6           | RICOH   | Aficio MP7001             | 48                 |
| P7           | RICOH   | imagio MP 7502            | 48                 |
| P8(Official) | Toshiba | e-studio 2051c-11606695   | 48                 |
| Total        |         |                           | 432                |

**Table 4**  
Brands and models of smartphones.

| No. | Brands | Models     |
|-----|--------|------------|
| S1  | Huawei | Mate40 Pro |
| S2  | Huawei | Nova5 Pro  |
| S3  | Iphone | X          |
| S4  | Iphone | 13         |
| S5  | Redmi  | K30        |

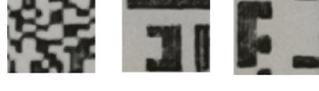
photograph the QR codes produced by each printer, as shown in Table 4, so 240 QR codes corresponding to each printer were collected.

Finally, a total of 2160 QR codes are collected, the image size of each collected QR code is unified to  $512 \times 512$ . In the field of printer source identification, the differences between categories are subtle, and the interference caused by image content can be avoided effectively through block processing, and the difference between categories can be amplified. Therefore, each QR code is divided into 64 patches, each of which is  $64 \times 64$  in size, and the number of QR code patches corresponding to each printer is 15,360, so there are 138,240 QR code patches in total. The examples and numbers of QR code patches produced by each printer are shown in Table 5.

As can be seen from Table 5, there are differences in the image texture of the QR codes produced by the different printers. The QR code image printed by the official printer is relatively regular, but the counterfeit process has undergone secondary printing. The quality of the counterfeit QR code pattern has declined to a certain extent, and the pattern printed by each counterfeit printer has different textures. Although the QR code pattern printed by Canon Image Runner Advance 6575 is clear, the pattern has a certain distortion, which is mainly caused by the geometric distortion of the printer. While the quality of the pattern printed by RICOH Imagio MP 7502 is distorted obviously, and even the pixels in the QR code pattern are seriously lost, which is caused by the loss of printing equipment and uneven toner diffusion. The above phenomena provide a basis for identifying the printer source of QR codes.

The QR code patch data are divided into training set, validation set and test set according to the ratio of 3:1:1, in which there are 82,944 image patches in the training set, 27,648 image patches in the validation set, and 27,648 image patches in the test set, as shown in Table 6.

**Table 5**  
Examples and number of QR code patches produced by each printer.

| The models of printers                     | Examples of QR code patches   | Number of image patches |
|--|---|-------------------------|
| Canon image Runner Advance 6575            |    | 15,360                  |
| Epson L15168                               |    | 15,360                  |
| Lenovo MD 7600                             |    | 15,360                  |
| RICOH Aficio MP 7502- I                    |    | 15,360                  |
| RICOH Aficio MP 7502- II                   |    | 15,360                  |
| RICOH Aficio MP 8001                       |   | 15,360                  |
| RICOH AficioMP 7001                        |  | 15,360                  |
| RICOH imagio MP 7502                       |  | 15,360                  |
| Toshiba e-studio 2051c-11606695 (Official) |  | 15,360                  |
| Total                                      |   | 138,240                 |

**Table 6**  
The data set distribution.

| Data set       | Number of samples |
|----------------|-------------------|
| Training set   | 82,944            |
| Validation set | 27,648            |
| Test set       | 27,648            |
| Total          | 138,240           |

#### 4.3. Ablation experiment of squeeze-excitation attention module

This section analyzes the effect of Squeeze-Excitation module on the performance of SE-BRB-Net through ablation experiment. The hyper-parameters of SE-BRB-Net are shown in Table 7.

The confusion matrix is used to measure the accuracy of the

**Table 7**

Hyper-parameters of the proposed SE-BRB-Net.

|                      |                     |                  |
|----------------------|---------------------|------------------|
| Learning rate: 0.001 | Momentum: 0.9       | Batch size:32    |
| Epochs: 30           | Weight decay:0.0001 | Solver step: SGD |

classification model : The true labels are represented by rows and the predicted labels are represented by columns, the correct predictions can be derived from the values on the diagonal, and a larger sum of all values on the diagonal generates a greater number of correct predictions, leading to a more effective model.

First, the SE module is removed from SE-BRB-Net and experiment is performed on the self-constructed data set, and the results are shown in Table 8. Then the experiment is conducted on the same data set using SE-BRB-Net, the hyper-parameter settings are the same for both sets of experiments, and the results are shown in Table 9.

From Tables 8 and 9, it can be seen that when the SE module is removed, the identification accuracy of SE-BRB-Net (without SE module) decreases on seven printers numbered P1, P2, P3, P4, P5, P6, P7, while there is only a small increase in the identification accuracy on the printers P0 and P8. In terms of overall identification accuracy, the accuracy of SE-BRB-Net (without SE module) is 98.45%, while the accuracy of SE-BRB-Net is 98.77%. It can be concluded that the addition of the SE module increases the identification accuracy of SE-BRB-Net by 0.32%, verifying the effectiveness of the SE module.

#### 4.4. Comparative experiments with different deep learning models

In the field of printer source identification, the commonly used CNNs include AlexNet (Krizhevsky et al., 2012) and GoogLeNet (Szegedy et al., 2015) and ResNet18 (He et al., 2016). In addition, there are CNNs proposed based on the characteristics of printer source identification, including PDSI proposed by Tsai (Tsai et al., 2019) and PSINet proposed by Guo (Guo et al., 2020): PDSI was designed in the order of  $5 \times 5$  conv1 layer, max pooling layer,  $5 \times 5$  conv2 layer, maxing pooling layer,  $5 \times 5$  conv3 layer, average pooling layer, fully connected layer and was mainly used for printer source identification of text documents collected by scanner and microscope, PSINet was designed in the order of  $5 \times 5$  convolutional layer, average pooling layer, BRB1, average pooling layer, BRB2, average pooling layer, BRB3, global average pooling layer, and a fully connected layer, the kernel sizes of all average pooling layers are  $5 \times 5$ , and PSINet was mainly used to identify the printer source of QR codes collected by scanner. Hence, this paper makes comparison with the above deep learning methods to verify the effectiveness and superiority of the proposed method, and the number of layers and parameters of each CNN are shown in Table 10. For PSINet, PDSI and the proposed SE-BRB-Net, the image size of the data set used in the experiments is  $64 \times 64$ . While for AlexNet, GoogleNet and ResNet18, the input image block size is standardized to  $227 \times 227$  or  $224 \times 224$  to meet the optimal performance requirements of these CNNs.

The confusion matrices of AlexNet, PSINet, PDSI, GoogleNet, ResNet18 and the proposed SE-BRB-Net are shown in Tables 11–16

**Table 8**

Confusion matrix of SE-BRB-Net (without SE module).

|            |    | Predicted (%) |              |              |              |              |             |              |              |            |
|------------|----|---------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|------------|
|            |    | P0            | P1           | P2           | P3           | P4           | P5          | P6           | P7           | P8         |
| Actual (%) | P0 | <b>99.25</b>  | 0            | 0.16         | 0.16         | 0.03         | 0.26        | 0.14         | 0            | 0          |
|            | P1 | 0.07          | <b>99.84</b> | 0.03         | 0            | 0            | 0.03        | 0            | 0.03         | 0          |
|            | P2 | 0.20          | 0            | <b>98.44</b> | 0.75         | 0.58         | 0.03        | 0            | 0            | 0          |
|            | P3 | 0.42          | 0            | 0.26         | <b>94.89</b> | 1.79         | 2.34        | 0.10         | 0.17         | 0.03       |
|            | P4 | 0.03          | 0            | 0.06         | 0.76         | <b>99.06</b> | 0.06        | 0            | 0            | 0.03       |
|            | P5 | 0.20          | 0.03         | 0.03         | 2.54         | 0.03         | <b>95.9</b> | 0.94         | 0.33         | 0          |
|            | P6 | 0.10          | 0            | 0            | 0.16         | 0            | 0.33        | <b>99.38</b> | 0.03         | 0          |
|            | P7 | 0             | 0            | 0            | 0.16         | 0            | 0.53        | 0.03         | <b>99.28</b> | 0          |
|            | P8 | 0             | 0            | 0            | 0            | 0            | 0           | 0            | 0            | <b>100</b> |

respectively.

It can be seen from Table 11 that the overall identification accuracy of AlexNet is 79.08%, and the performance on each printer is as follows: The identification accuracy on printers P1, P7 and P8 all exceeded 90%, and the printer P8 performed the best, with a recognition accuracy of 96.68%. The identification accuracy on printer P4 exceeds 80%, while the identification accuracy on the printers P0, P2, P5 and P6 is only above 70%. The identification accuracy on printer P3 is only 51.82%, which is the worst performance. The analysis found that the printer P3 is easily misclassified as printers P4 and P5, because printers P3, P4 and P5 belong to the same brand, and P3 and P4 belong to the same model, and the inherent features between printers of the same model are very similar, and the differences are relatively subtle. The size of the image input by AlexNet is large, and the inherent features that represent the properties of the printers cannot be extracted accurately. The convolution kernel of the first convolutional layer is  $11 \times 11$ , and large-scale convolution filters out the inherent features representing printer properties, resulting in unsatisfactory identification results.

It can be seen from Table 12 that the overall identification accuracy of PSINet is 91.87%, and the performance on each printer is as follows: PSINet performs best on printer P1, with a identification accuracy of 99.84%.The identification accuracy on printers P0, P2, P4, P6, P7 and P8 all exceed 90%, with 92.45%, 90.85%,95.35%, 94.82% and 99.74% respectively. The identification accuracy of PSINet on P5 ranged from 80% to 90%, at 82.91%. The identification accuracy on P3 only exceeded 70%, at 76.4%. PSINet is mainly designed for printer source identification of QR code images scanned by the scanner, the scanned QR code image quality is high and clear, so PSINet with simple structure can achieve desired identification result. However, the scanner is specialised equipment and have problems such as large size and inconvenience to carry, making it difficult to be widely used. Therefore, this paper proposes printer source identification for QR codes captured by smart-phones. In this case, the QR code image are affected by light changes and ambient noise, making it difficult to effectively extract the subtle features representing printer attributes, resulting in a significant degradation in the performance of PSINet.

It can be seen from Table 13 that the overall identification accuracy of PDSI is 92.18%, and the performance on each printer is as follows: The identification accuracy on printers P0, P1 and P8 all exceeded 99%, with the highest accuracy achieved on printer P8 at 99.48%. The identification accuracy on printers P2, P4, P6 and P7 all exceeded 90%, with 90.72%, 93.75%, 93.95% and 96.42%, respectively. The identification accuracy on printer P5 is 83.89%, and the worst performance is on printer P3, which has an identification accuracy of 78.81%. Although PDSI has only 4 layers, the overall identification accuracy is 13.1% higher than that of AlexNet. This is because the image input size of PDSI is small, which avoids the interference caused by the image content in the process of subtle feature extraction. Hence, it is easier to accurately extract the inherent features of the printer, and the convolution kernel size of the first convolutional layer is small, which can more accurately extract subtle feature differences representing printer attributes.

We see from Table .14 that the overall identification accuracy of

**Table 9**  
Confusion matrix of SE-BRB-Net.

| Avg 98.77% |    | Predicted (%) |            |              |              |              |              |              |              |              |
|------------|----|---------------|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            |    | P0            | P1         | P2           | P3           | P4           | P5           | P6           | P7           | P8           |
| Actual (%) | P0 | <b>99.12</b>  | 0          | 0.2          | 0.36         | 0            | 0.29         | 0.03         | 0            | 0            |
|            | P1 | 0             | <b>100</b> | 0            | 0            | 0            | 0            | 0            | 0            | 0            |
|            | P2 | 0.16          | 0          | <b>98.96</b> | 0.49         | 0.39         | 0            | 0            | 0            | 0            |
|            | P3 | 0.46          | 0          | 0.29         | <b>95.74</b> | 1.63         | 1.79         | 0.06         | 0.03         | 0            |
|            | P4 | 0.1           | 0          | 0.06         | 0.52         | <b>99.32</b> | 0            | 0            | 0            | 0            |
|            | P5 | 0.36          | 0          | 0.03         | 1.79         | 0.06         | <b>96.65</b> | 0.85         | 0.2          | 0.06         |
|            | P6 | 0.1           | 0          | 0            | 0.03         | 0            | 0.13         | <b>99.74</b> | 0            | 0            |
|            | P7 | 0             | 0.03       | 0            | 0.13         | 0            | 0.32         | 0            | <b>99.41</b> | 0            |
|            | P8 | 0             | 0          | 0            | 0            | 0            | 0.03         | 0            | 0            | <b>99.97</b> |

**Table 10**  
Comparison of number of layers and parameters of six CNN models.

| CNNs                              | Number of layers | Number of parameters |
|-----------------------------------|------------------|----------------------|
| AlexNet (Krizhevsky et al., 2012) | 8                | 56,881,865           |
| PSINet (Guo et al., 2020)         | 11               | 218,825              |
| PDSI (Tsai et al., 2019)          | 4                | 114,601              |
| GoogLeNet (Szegedy et al., 2015)  | 22               | 5,976,505            |
| ResNet18 (He et al., 2016)        | 18               | 11,202,633           |
| <b>SE-BRB-Net</b>                 | <b>14</b>        | <b>664,729</b>       |

GoogLeNet is 89.4%, and the performance on each printer is as follows. The identification accuracy on printers P1, P7 and P8 all exceed 99%, with the highest accuracy of 99.8% on printers P7 and P8. The identification accuracy on P2 is 93.29%; the identification accuracy on P0, P4, P5 and P6 is between 80% and 90%; the identification accuracy on printer P3 is the lowest with only 63.64%. GoogLeNet is 10.32% higher than AlexNet in terms of accuracy due to the deeper layers and the stronger multi-scale feature representation ability of the Inception module, but it is still 2.78% lower than PSDI. This is because GoogLeNet does not use small-size image blocks for training.

It can be seen from Table 15 that the overall identification accuracy of ResNet18 is 97.6%, and the performance on each printer is as follows. The identification accuracy on printers P0, P1, P4, P6, P8 and P9 all

**Table 11**  
Confusion matrix of AlexNet.

| Avg 79.08% |    | Predicted (%) |              |              |              |              |              |              |              |              |
|------------|----|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            |    | P0            | P1           | P2           | P3           | P4           | P5           | P6           | P7           | P8           |
| Actual (%) | P0 | <b>71.81</b>  | 0            | 6.15         | 8.95         | 1.60         | 8.53         | 2.96         | 0            | 0            |
|            | P1 | 0.03          | <b>92.61</b> | 0.33         | 0.33         | 0            | 0.36         | 0.52         | 5.7          | 0.12         |
|            | P2 | 9.83          | 0.07         | <b>72.17</b> | 10.64        | 4.65         | 1.46         | 0.88         | 0.23         | 0.07         |
|            | P3 | 4.43          | 0            | 2.02         | <b>51.82</b> | 16.86        | 22.4         | 1.01         | 0.81         | 0.65         |
|            | P4 | 1.76          | 0            | 0.85         | 9.83         | <b>84.08</b> | 3.19         | 0.16         | 0            | 0.13         |
|            | P5 | 0.88          | 0.16         | 0.23         | 4.85         | 1.56         | <b>74.61</b> | 7.52         | 6.45         | 3.65         |
|            | P6 | 0.78          | 0            | 0.23         | 0.59         | 0.03         | 12.89        | <b>74.87</b> | 10.09        | 0.52         |
|            | P7 | 0             | 0.36         | 0            | 0.1          | 0            | 2.38         | 1.88         | <b>93.03</b> | 2.25         |
|            | P8 | 0             | 0            | 0            | 0.07         | 1.17         | 1.52         | 0.46         | 0.1          | <b>96.68</b> |

**Table 12**  
Confusion matrix of PSINet.

| Avg 91.87% |    | Predicted (%) |              |              |             |              |              |              |              |              |
|------------|----|---------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|--------------|
|            |    | P0            | P1           | P2           | P3          | P4           | P5           | P6           | P7           | P8           |
| Actual (%) | P0 | <b>92.45</b>  | 0            | 1.60         | 3.91        | 0.13         | 1.56         | 0.32         | 0.03         | 0            |
|            | P1 | 0.07          | <b>99.84</b> | 0            | 0           | 0.03         | 0            | 0            | 0.06         | 0            |
|            | P2 | 1.6           | 0.06         | <b>90.85</b> | 4.43        | 2.64         | 0.29         | 0.03         | 0.1          | 0            |
|            | P3 | 2.96          | 0            | 3.81         | <b>76.4</b> | 7.91         | 7.68         | 0.29         | 0.59         | 0.36         |
|            | P4 | 0.26          | 0            | 1.30         | 2.38        | <b>95.35</b> | 0.39         | 0            | 0            | 0.32         |
|            | P5 | 1.82          | 0            | 0.23         | 6.84        | 0.88         | <b>82.91</b> | 4.13         | 2.6          | 0.59         |
|            | P6 | 0.68          | 0.06         | 0            | 0.36        | 0            | 3.65         | <b>94.82</b> | 0.33         | 0.1          |
|            | P7 | 0.03          | 1.3          | 0.23         | 0.13        | 0.03         | 2.6          | 0.69         | <b>94.47</b> | 0.52         |
|            | P8 | 0             | 0.03         | 0            | 0           | 0.13         | 0.1          | 0            | 0            | <b>99.74</b> |

exceeded 98%, with the highest identification accuracy of 99.93% on printers P1 and P8. The identification accuracy on printers P2, P3 and P5 all exceeded 90%, with 97.07%, 92.29% and 94.3%, respectively. The above results verify the advantages of the residual block in extracting effective subtle features of printer attributes.

As can be seen from Table 16, the overall identification accuracy of proposed SE-BRB-Net reached 98.77%, in which the identification accuracy of P1 reached 100%, the identification accuracy of printer P8 also reached 99.97%, the identification accuracy on printers P0, P4, P6 and P7 also exceeded 99%, with 99.12%, 99.32%, 99.74% and 99.41% respectively. The accuracy on printer P2 is 98.96%, and the identification accuracy on printer P5 and P3 is 96.65% and 95.54% respectively. The proposed SE-BRB-Net uses small-size image patches for training, which can better highlight the subtle features of printers. And the SE-BRB module makes improvements on the bottleneck residual block with strong subtle feature extraction capability, leading to the best identification results.

For a more intuitive comparison of several convolutional neural networks, the identification accuracy is shown in Table 17 and Fig. 7. The analysis is as follows:

- (1) The average identification accuracy of AlexNet is 79.08%, the average identification accuracy of PSINet is 91.87%, the average

**Table 13**  
Confusion matrix of PDSI.

| Avg 92.18% |    | Predicted (%) |              |              |              |              |              |              |              |              |
|------------|----|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            |    | P0            | P1           | P2           | P3           | P4           | P5           | P6           | P7           | P8           |
| Actual (%) | P0 | <b>93.33</b>  | 0            | 1.56         | 3.16         | 0.23         | 1.36         | 0.36         | 0            | 0            |
|            | P1 | 0             | <b>99.32</b> | 0.26         | 0.23         | 0            | 0.03         | 0.03         | 0.13         | 0            |
|            | P2 | 2.34          | 0.23         | <b>90.72</b> | 4.39         | 1.96         | 0.26         | 0            | 0.1          | 0            |
|            | P3 | 2.18          | 0.03         | 3.84         | <b>78.81</b> | 7.06         | 6.74         | 0.2          | 0.78         | 0.36         |
|            | P4 | 0.1           | 0.26         | 1.63         | 3.94         | <b>93.75</b> | 0.03         | 0            | 0.03         | 0.26         |
|            | P5 | 0.94          | 0.1          | 0.36         | 7.02         | 0.07         | <b>83.89</b> | 5.21         | 1.56         | 0.85         |
|            | P6 | 0.28          | 0            | 0.1          | 0.36         | 0            | 4.43         | <b>93.95</b> | 0.78         | 0.1          |
|            | P7 | 0.1           | 0.16         | 0.06         | 0.06         | 0            | 2.25         | 0.62         | <b>96.42</b> | 0.33         |
|            | P8 | 0             | 0            | 0            | 0.04         | 0.04         | 0.3          | 0.04         | 0.1          | <b>99.48</b> |

**Table 14**  
Confusion matrix of GoogLeNet.

| Avg 89.4%  |    | Predicted (%) |              |              |              |              |              |              |             |             |
|------------|----|---------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|
|            |    | P0            | P1           | P2           | P3           | P4           | P5           | P6           | P7          | P8          |
| Actual (%) | P0 | <b>89.91</b>  | 0            | 2.67         | 2.08         | 0.29         | 4.07         | 0.98         | 0           | 0           |
|            | P1 | 0             | <b>98.73</b> | 0            | 0            | 0            | 0            | 0            | 1.24        | 0.03        |
|            | P2 | 0.72          | 0.03         | <b>93.29</b> | 4.04         | 0.72         | 0.45         | 0.43         | 0.32        | 0           |
|            | P3 | 0.39          | 0            | 1.98         | <b>63.64</b> | 4.07         | 26.40        | 0.59         | 2.77        | 0.16        |
|            | P4 | 0.36          | 0            | 0.5          | 7.1          | <b>86.36</b> | 5.50         | 0.03         | 0.06        | 0.09        |
|            | P5 | 0             | 0.03         | 0.26         | 1.43         | 0.29         | <b>88.74</b> | 0.88         | 7.55        | 0.81        |
|            | P6 | 0             | 0            | 0.06         | 0.85         | 0            | 6.16         | <b>84.34</b> | 8.43        | 0.16        |
|            | P7 | 0             | 0            | 0            | 0            | 0            | 0.17         | 0            | <b>99.8</b> | 0.03        |
|            | P8 | 0             | 0            | 0            | 0            | 0.03         | 0.14         | 0.03         | 0           | <b>99.8</b> |

**Table 15**  
Confusion matrix of ResNet18.

| Avg 97.6%  |    | Predicted (%) |              |              |              |             |             |             |              |              |
|------------|----|---------------|--------------|--------------|--------------|-------------|-------------|-------------|--------------|--------------|
|            |    | P0            | P1           | P2           | P3           | P4          | P5          | P6          | P7           | P8           |
| Actual (%) | P0 | <b>98.6</b>   | 0            | 0.23         | 0.55         | 0.03        | 0.39        | 0.2         | 0            | 0            |
|            | P1 | 0.03          | <b>99.93</b> | 0            | 0            | 0.04        | 0           | 0           | 0            | 0            |
|            | P2 | 0.32          | 0            | <b>97.07</b> | 1.76         | 0.72        | 0.10        | 0           | 0            | 0.03         |
|            | P3 | 0.62          | 0            | 0.88         | <b>92.29</b> | 2.41        | 3.42        | 0.16        | 0.19         | 0.03         |
|            | P4 | 0.03          | 0            | 0.33         | 1.01         | <b>98.6</b> | 0           | 0           | 0            | 0.03         |
|            | P5 | 0.42          | 0            | 0.03         | 3.22         | 0.1         | <b>94.3</b> | 1.44        | 0.39         | 0.1          |
|            | P6 | 0.06          | 0            | 0            | 0.06         | 0           | 1.05        | <b>98.8</b> | 0.03         | 0            |
|            | P7 | 0             | 0            | 0            | 0.2          | 0           | 0.62        | 0           | <b>99.15</b> | 0.03         |
|            | P8 | 0             | 0            | 0            | 0            | 0           | 0.07        | 0           | 0            | <b>99.93</b> |

**Table 16**  
Confusion matrix of SE-BRB-Net.

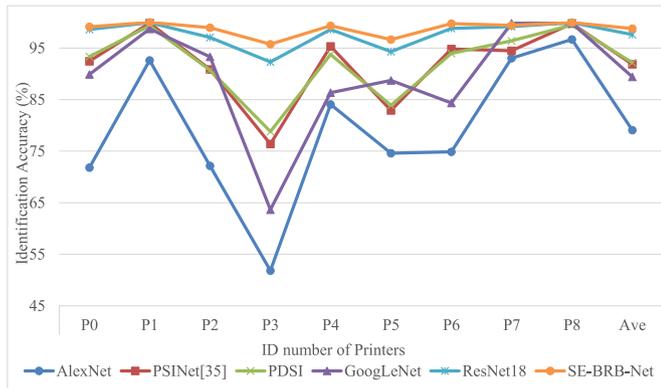
| Avg 98.77% |    | Predicted (%) |            |              |              |              |              |              |              |              |
|------------|----|---------------|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            |    | P0            | P1         | P2           | P3           | P4           | P5           | P6           | P7           | P8           |
| Actual (%) | P0 | <b>99.12</b>  | 0          | 0.2          | 0.36         | 0            | 0.29         | 0.03         | 0            | 0            |
|            | P1 | 0             | <b>100</b> | 0            | 0            | 0            | 0            | 0            | 0            | 0            |
|            | P2 | 0.16          | 0          | <b>98.96</b> | 0.49         | 0.39         | 0            | 0            | 0            | 0            |
|            | P3 | 0.46          | 0          | 0.29         | <b>95.74</b> | 1.63         | 1.79         | 0.06         | 0.03         | 0            |
|            | P4 | 0.1           | 0          | 0.06         | 0.52         | <b>99.32</b> | 0            | 0            | 0            | 0            |
|            | P5 | 0.36          | 0          | 0.03         | 1.79         | 0.06         | <b>96.65</b> | 0.85         | 0.2          | 0.06         |
|            | P6 | 0.1           | 0          | 0            | 0.03         | 0            | 0.13         | <b>99.74</b> | 0            | 0            |
|            | P7 | 0             | 0.03       | 0            | 0.13         | 0            | 0.32         | 0            | <b>99.41</b> | 0            |
|            | P8 | 0             | 0          | 0            | 0            | 0            | 0.03         | 0            | 0            | <b>99.97</b> |

identification accuracy of PDSI is 92.18%, the average identification accuracy of GoogLeNet is 89.4%, and the average identification accuracy of ResNet18 is 97.6%, and the average identification accuracy of SE-BRB-Net is 98.77%. The average accuracy of SE-BRB-Net is 19.69%, 6.9%, 6.59%, 9.37% and 1.17% higher than AlexNet, PSINet, PDSI, GoogLeNet and ResNet18, respectively.

(2) The proposed SE-BRB-Net achieves the highest accuracy on printers P0, P1, P2, P3, P4, P5, P6, P8, where the identification accuracy reaches 100% on printer P1, and the identification accuracy on the printer P8 reached 99.97%. Compared with other five CNNs, the proposed SE-BRB-Net can also effectively distinguish printers of the same brand such as P3, P4, P5 and P6, showing the superiority of the Squeeze-Excitation attention module in extracting features that represent printer attributes.

**Table 17**  
Identification accuracy of six CNNs.

| CNNs                              | Identification Accuracy (%) |            |              |              |              |              |              |             |              |              |
|-----------------------------------|-----------------------------|------------|--------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|
|                                   | P0                          | P1         | P2           | P3           | P4           | P5           | P6           | P7          | P8           | Ave          |
| AlexNet (Krizhevsky et al., 2012) | 71.81                       | 92.61      | 72.17        | 51.82        | 84.08        | 74.61        | 74.87        | 93.03       | 96.68        | 79.08        |
| PSINet (Guo et al., 2020)         | 92.45                       | 99.84      | 90.85        | 76.4         | 95.35        | 82.91        | 94.82        | 94.47       | 99.74        | 91.87        |
| PDSI (Tsai et al., 2019)          | 93.33                       | 99.32      | 90.72        | 78.81        | 93.75        | 83.89        | 93.95        | 96.42       | 99.48        | 92.18        |
| GoogLeNet (Szegedy et al., 2015)  | 89.91                       | 98.73      | 93.29        | 63.64        | 86.36        | 88.74        | 84.34        | <b>99.8</b> | 99.8         | 89.4         |
| ResNet18 (He et al., 2016)        | 98.6                        | 99.93      | 97.07        | 92.29        | 98.6         | 94.3         | 98.8         | 99.15       | 99.93        | 97.6         |
| SE-BRB-Net                        | <b>99.12</b>                | <b>100</b> | <b>98.96</b> | <b>95.74</b> | <b>99.32</b> | <b>96.65</b> | <b>99.74</b> | 99.41       | <b>99.97</b> | <b>98.77</b> |



**Fig. 7.** Identification accuracy of six CNNs.

In addition to identification accuracy, training time and inference time are also common metrics to measure CNN's performance. The faster the training and inference speed, the better the performance of the CNN model. The training time and inference time of the six CNNs are shown in Table 18, where h, m, s represent hour, minute and second, respectively.

In terms of training time, the PDSI has the least training time of 17 min and 12 s due to the fact that it has only three convolutional layers and one fully connected layer. It comes PSINet, which has 11 layers and consists mainly of bottleneck residual blocks, and has only one fully connected layer, which greatly reduces the number of parameters, and hence the training time is also less, at 58 min and 55 s. Next is the 8-layer AlexNet with a training time of 2 h and 2 min, AlexNet uses  $11 \times 11$  sized convolutional layer and has three fully connected layers, which greatly increases its number of parameters and training time. The 14-layer SE-BRB-Net proposed in this paper performs moderately well, with a training time of 2 h and 41 min, it has more layers than the previous three networks, and has a squeeze-excitation module embedded in each bottleneck residual block, which increases the training time to some extent. While GoogLeNet and ResNet18 have longer training time due to more layers, which are 5 h and 48 min and 7 h and 8 min, respectively. It can be concluded that the training time of CNN is closely related to its structure. The fewer layers, parameters and fully connected layers of CNN, the shorter the training time is generally.

In terms of inference time, PDSI also takes the least time, at 0.45 s; then PSINet, at 0.49 s, followed by SE-BRB-Net, at 0.52 s; the inference speeds of the above three are relatively close. In contrast, the inference

**Table 18**  
The training time and inference time of the six CNNs.

| CNNs                              | Training time | Inference time (s/QR code) |
|-----------------------------------|---------------|----------------------------|
| AlexNet (Krizhevsky et al., 2012) | 2h2m          | 0.93                       |
| PSINet (Guo et al., 2020)         | 58m55s        | 0.49                       |
| PDSI (Tsai et al., 2019)          | 17m12s        | 0.45                       |
| GoogLeNet (Szegedy et al., 2015)  | 5h48 m        | 2.31                       |
| ResNet18 (He et al., 2016)        | 7h8 m         | 1.75                       |
| SE-BRB-Net                        | 2h41 m        | 0.52                       |

of AlexNet, ResNet18 and GoogLeNet takes longer, 0.93 s, 1.75 s and 2.31 s respectively. It can be inferred that the inference speed is closely related to the structure of the model and the number of parameters, reducing the number of parameters will reduce the inference time, so designing lightweight model will be considered in further research. In addition, the inference speed is also affected by the performance of the computing platform, including memory access costs, hardware characteristics, software implementation, system environment, etc. In subsequent research, we will test the proposed model in the actual deployment environment.

## 5. Conclusion and future works

The QR code images captured by smartphones suffer from the issues such as reduced clarity and blurring, causing the existing deep learning-based printing source identification network to be unsatisfactory. In order to solve the above problem, this paper proposes a convolutional neural network named SE-BRB-Net that combines the bottleneck residual block and the squeeze-excitation attention module. The network has 14 layers, with the squeeze-excitation bottleneck residual block as the basic unit. By incorporating a squeeze-excitation block in each bottleneck residual block, the model of SE-BRB-Net pays more attention to the information that is useful for printer source identification and attenuates the interference of the useless information to achieve a better feature extraction capability of the printer attributes.

To address the problem of QR code printer source identification under smartphone collection conditions, nine printer are first used to print the QR code digital images, and then five smartphones of different brands or models are used to collect the QR code labels. The experimental results show that the average identification accuracy of the SE-BRB-Net model on nine printers reaches 98.77%, which is a relatively superior performance compared with other deep learning networks, and significantly improves the identification accuracy of the model under the premise of controlling the depth of the network, the number of parameters, and the inference time. Thus, the model complexity and the identification accuracy is therefore balanced well, which fully demonstrates that the proposed SE-BRB-Net is superior in solving the problem of printer source identification of QR codes under smartphone capture conditions. It is interesting to note that the deep learning approach proposed in this paper can also be applied to the authentication of text documents to support judicial forensics.

Future research will focus on the following aspects:

- (1) More and larger data sets are established to test the performance of the proposed algorithm.
- (2) We will try to design lightweight model with the aim of ensuring identification performance while reducing the number of parameters and model complexity.
- (3) Since there are so many brands, models and numbers of printers in the real world, and new printers are emerging all the time, we will investigate effective solutions to quickly identify new printers, such as meta-learning based methods.

## CRedit authorship contribution statement

**Zhongyuan Guo:** Conceptualization, Methodology, Software, Validation, Visualization, Data curation, Writing – original draft. **Shiyuan Wang:** Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Zhaohui Zheng:** Investigation, Formal analysis, Writing – review & editing. **Ke Sun:** Resources, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This study is supported in part by the National Natural Science Foundation of China (Grant No. 62071391, 62306245), the Science Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN202300225), Chongqing Postdoctoral International Exchange and Training Program, Chongqing Postdoctoral Science Foundation Special Funded (Grant No. 2022CQBSHTB3076).

## References

- Agarap, A.F., 2018. Deep Learning Using Rectified Linear Units (Relu) [J] arXiv preprint arXiv:1803.08375.
- Aini, Q., Rahardja, U., Tangkaw, M.R., et al., 2020. Embedding a blockchain technology pattern into the QR code for an authentication certificate. *Journal Online Informatika* 5 (2), 239–244.
- Ali, G.N., Mikkilineni, A.K., Allebach, J.P., et al., 2003. Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices[C]// NIP & digital fabrication conference, 2003 (2), 511–515.
- Bibi, M., Hamid, A., Moetesum, M., et al., 2019. Document forgery detection using printer source identification—a text-independent approach[C]//2019 international conference on document analysis and recognition workshops (ICDARW). *IEEE* 8, 7–12.
- Chan, T.H., Jia, K., Gao, S., et al., 2015. PCANet: a simple deep learning baseline for image classification. *IEEE Trans. Image Process.* 24 (12), 5017–5032.
- Chen, C., Li, M., Ferreira, A., et al., 2019. A copy-proof scheme based on the spectral and spatial barcoding channel models. *IEEE Trans. Inf. Forensics Secur.* 15, 1056–1071.
- Choi, J.H., Lee, H.Y., Lee, H.K., 2013. Color laser printer forensic based on noisy feature and support vector machine classifier. *Multimed. Tool. Appl.* 67 (2), 363–382.
- Druzhdov, P.N., Kustikova, V.D., 2016. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recogn. Image Anal.* 26 (1), 9–15.
- Ferreira, Anselmo, Bondi, L., Baroffio, L., et al., 2017. Data-driven feature characterization techniques for laser printer attribution. *IEEE Trans. Inf. Forensics Secur.* 12 (8), 1860–1873.
- Focardi, R., Luccio, F.L., Wahsheh, H.A.M., 2019. Useable security for QR code. *J. Inf. Secur. Appl.* 48, 102369.
- Ghosh, S., Das, N., Das, I., et al., 2019. Understanding deep learning techniques for image segmentation. *ACM Comput. Surv.* 52 (4), 1–35.
- Gu, J., Wang, Z., Kuen, J., et al., 2018. Recent advances in convolutional neural networks. *Pattern Recogn.* 77, 354–377.
- Guo, Z., Zheng, H., You, C., Xu, X., Wu, X., Zheng, Z., Ju, J., 2020. Digital forensics of scanned QR code images for printer source identification using bottleneck residual block. *Sensors* 20 (21), 6305, 6018.
- Hamzehyan, R., Razzazi, F., Behrad, A., 2021. Printer source identification by feature modeling in the total variable printer space. *J. Forensic Sci.* 66 (6), 2261–2273.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770–778.
- He, T., Zhang, Z., Zhang, H., et al., 2019. Bag of Tricks for Image Classification with Convolutional Neural networks[C]. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 558–567.
- Hu, J., Chen, L., Sun, G., 2018. Squeeze-and-excitation networks[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141.
- Jain, H., Joshi, S., Gupta, G., et al., 2020. Passive classification of source printer using text-line-level geometric distortion signatures from scanned images of printed documents. *Multimed. Tool. Appl.* 79 (11), 7377–7400.
- Jiang, B., Chen, S., Wang, B., et al., 2022. MGLNN: semi-supervised learning via multiple graph cooperative learning neural networks. *Neural Network.* 153, 204–214.
- Khan, A., Sohail, A., Zahoor, U., et al., 2020. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* 53, 5455–5516.
- Kim, Do-Guk, Hou, Jong-Uk, Lee, Heung-Kyu, 2019. Learning deep features for source color laser printer identification based on cascaded learning. *Neurocomputing* 365, 219–228.
- Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25 (2), 1097–1105.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521 (7553), 436–444.
- Mikkilineni, A.K., Khanna, N., Delp, E.J., 2010. Texture based attacks on intrinsic signature based printer identification[C]//Conference on media forensics and security II(CMFS II). *SPIE* 7541, 295–306.
- Minaee, S., Boykov, Y., Porikli, F., et al., 2021. Image segmentation using deep learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (7), 3523–3542.
- Nguyen, H.P., Retraint, F., Morain-Nicolier, F., et al., 2019. A watermarking technique to secure printed matrix barcode—application for anti-counterfeit packaging. *IEEE Access* 7, 131839–131850.
- Oliver, J., Chen, J., 2002. Use of signature analysis to discriminate digital printing technologies[C]//2002. *International Conference on Digital Printing Technologies (NIP) 2002* (1), 218–222.
- Roy, A.M., Bhaduri, J., 2023. DenseSPH-YOLOv5: an automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism. *Adv. Eng. Inf.* 56, 102007–102022.
- Szegedy, C., Liu, W., Jia, Y., et al., 2015. Going Deeper with Convolutions[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.
- Theodoridis, S., 2015. *Neural Networks and Deep Learning [J]. Machine Learning*, pp. 875–936.
- Tsai, M.J., Hsu, C.L., Yin, J.S., et al., 2016. Digital Forensics for Printed Character Source Identification[C]. 2016 *IEEE International Conference on Multimedia and Expo (ICME)*. *IEEE*, pp. 1–6.
- Tsai, Min-Jen, Tao, Yu-Han, Imam, Yuadi, 2019. Deep learning for printed document source identification. *Signal Process. Image Commun.* (70), 184–198.
- Wang, G., Li, W., Zuluaga, M.A., et al., 2018. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Trans. Med. Imag.* 37 (7), 1562–1573.
- Zhao, Z.Q., Zheng, P., Xu, S., et al., 2019. Object detection with deep learning: a review. *IEEE Transact. Neural Networks Learn. Syst.* 30 (11), 3212–3232.
- Zhou, Q., Yan, Y., Fang, T., et al., 2016. Text-independent printer identification based on texture synthesis. *Multimed. Tool. Appl.* 75 (10), 5557–5580.