A Novel Electronic Component Classification Algorithm Based on Hierarchical Convolution Neural Network

Xiaomei Hu¹, Jun Xu¹* and Jun Wu¹

¹Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China

*Corresponding author’s e-mail: Rocky554763131@163.com

Abstract. In the paper, the author proposes a recognition and classification algorithm of electronic components based on hierarchical convolutional neural network (NH-CNN) that reduces the computational complexity on the drawbacks of traditional image classification method based on deep learning such as the failure of effectively combining multiple deep characteristics, the poor performance of classifier, the difficulty of parameter adjustment and the long training time. The algorithm is trained through the Convolutional Automatic Coding (CAE) layer to obtain relevant feature maps, which reduces the input parameters of the designed convolutional neural network. Since the CAE model and the convolutional neural network have similar convolution and pooling operations, the feature maps obtained from the CAE model are put into the designed neural network based on the transfer learning method. Finally, the feature fusion method is used to output the obtained features to the fully connected layer, which is used to better express the depth information contained in the electronic component images and improve the accuracy of classification. The experimental results show that the proposed algorithm can effectively extract depth features with high precision of 94.26% and less complexity and overcome the defects of traditional image classification algorithms such as manual image extraction and low classification efficiency.

1. Introduction

Electronic components play an important role in the development of industry, because they are various in types and shapes, and evolve towards miniaturization and chip. And in its production, scientific research, application and recycling, the classification is a very important basic work, so it is of great practical significance to design an automatic identification system of electronic components that can be processed in real time.

The current image classification methods are mainly divided into two categories. The first is based on the image space domain or the transform domain to classify images and the second is to automatically learn image features for image classification by use of convolutional neural networks (CNN) [1]. In recent years, CNN has been widely used in image recognition detection neighborhoods and has achieved fruitful results. In particular, Regions with CNN (R-CNN) [2], Fast R-CNN [3] network model and Faster R-CNN [4] network model proposed in recent years have achieved good detection performance in the field of target recognition. However, whether the performance is good or bad depends on the expression of the characteristic information and the adjustment of the training parameters. Therefore, this paper proposes an electronic component recognition and classification method based on hierarchical convolutional neural network model, which aims to improve the effect
of information expression of high level features in the CNN models and reduce the number of parameters in training while directly performing automatic feature learning on the input image so as to make up for the shortcomings of the traditional electronic component classification method.

2. Algorithm principle

2.1. Convolutional Auto Encoding (CAE) layer

Assume that the input image is $x$. For the convenience of the development, assuming CAEs are utilized for image classification tasks. For each image $x \in \mathbb{R}^{w \times h \times c}$, $w$, $h$, $c$ refer to the image width, height, and number of channels respectively. Figure 1 illustrates the architecture of one CAE.

![Figure 1. Convolutional Auto Encoding (CAE) layer](image)

As shown in Figure 1, a CAE is composed of a convolutional layer followed by a pooling layer and then a deconvolutional layer. The transformation through the convolutional layer and then the pooling layer is called the encoder. The transformation through the deconvolutional layer is called the decoder.

2.1.1. Image preprocessing

Before the target recognition of the electronic components, the image acquisition of the electronic components is first performed, and then the acquired images are simply preprocessed.

(1) Image denoising: Since there will be some noise in the image in natural scenes, in order to enhance the three-dimensional robustness of the image, noise can be added during image preprocessing to perform network brightening and denoising training. In addition to the common Gaussian noise, this framework also adds salt and pepper noise, poisson noise, and speckle noise to enhance the robustness of the network.

(2) Gamma correction: Because Gamma correction is performed at the pixel level, it does not bring discontinuous spots to the image, and it is not necessary to consider the position of the image light source to adjust the image [5], so Gamma can be used in the image preprocessing stage Correction. The essence of Gamma correction is to adjust the Gamma curve of the image, which is a non-linear operator often used in the image preprocessing stage. The image adjustment formula is as follows.

$$x_d = A \times x^\gamma$$  \hspace{1cm} (1)

Where: $A$ is a constant, $\gamma$ is a variable, when $\gamma > 1$, it can reduce the contrast of the image and achieve the effect of low light in the image; when $\gamma < 1$, the contrast of the image can be increased to achieve the effect of image highlights; when $\gamma = 1$, it has no effect on the image. After getting the preprocessed image $x_d$, $x_d$ could be input into the CAE network for training.
2.1.2. **convolutional layer**

The convolutional layer is the feature extraction layer, where the convolution kernel contained in the convolutional layer is the core of neural network training. Each different convolution kernel can extract different input data features, and the more the number of convolution kernels, the more features are extracted. The relationship between the feature image and the expression is as follows:

\[ h(x) = \sigma(x_d \ast W^k + b^k) \]  

Where: \( h(x) \) represents a function which can obtain \( k \) feature maps though convolutional layer, \( x_d \) represents the preprocessed image, \( \sigma \) represents the activation function, \( W^k \) represents \( k \) convolutional kernels (in Figure 1, \( k = 3 \)), and the convolutional calculation is represented by \( \ast \), \( b^k \) represents the additive bias term.

2.1.3. **pooling layer**

Pooling layer (also called the down sampling layer). The pooling layer is usually accessed after the convolutional layer. The pooling layer can perform feature aggregation statistics on the local position of the image, reducing the working dimension of the feature vector, reducing the amount of data, and speeding up the network training speed while retaining valid data. Its input and output can be expressed by equation (3):

\[ H(x) = down(max(h(x))) \]  

2.1.4. **deconvolutional layer**

The pooling operation is the last step in the encoding phase. After completion, the network starts the decoding process. During the decoding phase, a deconvolution operation can be performed to reconstruct the image. Because the size and step size of the unreasonable convolution kernel may cause a checkerboard effect on the deconvolution operation, that is, the problem of overlapping matrix blocks may occur. In this framework, upsampling and convolution are used to perform image information recovery.

The upsampling operation has restored the size of the feature map, so the convolution operation at the decoding stage does not need to consider the size of the image, and only a simple convolution operation is required. The specific convolution operation formula is as follows:

\[ Y(x) = \sigma(y \ast w^k + b'^k) \]  

Where: \( Y(x) \) represents the deconvolutional function, \( y \) represents a feature map which equals to the original image size, \( w^k \) represents \( k \) convolutional kernels, and the convolutional calculation is represented by \( \ast \), \( b'^k \) represents the additive bias term in the deconvolutional layer.

2.2. **Convolutional Neural Network (CNN) Training Layer**

The key idea of our method is to train a specialized CNN structure to extract robust hierarchical features from image patches and provide them to softmax classifiers.

The network consists of three parts: the first part is a combination of the convolutional layer and the polling layer as the basic module of the deep network; the second part is the fully connected layer; the third part is the classifier part, that is, the high-level abstract features extracted after the fully connected layers are put into the softmax classifier. Finally, the classification results are obtained. The network structure is as shown in Figure 2.
Algorithm 1 Framework of the proposed CNN Algorithm

**Input:** Original Image \( x \in \mathbb{R}^{w \times h \times c} \)

**Output:** \( H_1, H_2, \ldots, H_p \)

**Begin:**
- Preprocessing Image \( x_d = A \times x^\gamma \)
- Convolutional feature map \( h_1 = \text{convolution} \ x_d \) by Eq.(2);
- Pooling feature map \( H_1 = \text{down} \ h_1 \) by Eq.(3);
- Upsample feature map \( y_1 \);
- Reconstructed image \( Y_1 = \text{convolution} \ y \) by Eq.(4);

**For** (\( e \leqslant \text{epoch} \)) do:
  - Pooling feature map \( H_i (i=1,2,3,\ldots) \);
  - Convolutional feature map \( h_{i+1} = \text{convolution} \ H_i \);
  - Pooling feature map \( H_{i+1} = \text{down} \ h_{i+1} \);
  - Upsample feature map \( y_i \);
  - Reconstructed image \( Y_i = \text{convolution} \ y_i \);
  - Parameter update;

**End.**

Algorithm 1 outlines the framework of the CNN algorithm. The first step of the algorithm is to initialize the convolution layer feature \( h_1 \) and the pooling layer feature \( H_1 \) based on the transfer learning model; the second step is to obtain the convolution layer feature \( h_2 \) according to the transfer learning model and the pooling layer feature \( H_1 \) in the first step and pooling layer characteristics \( H_2 \); by performing this operation, different convolution layer feature \( h_1, h_2, \ldots, h_p \) and pooling layer characteristics \( H_1, H_2, \ldots, H_p \) are obtained.
2.3. **Full Connection Layer**

The fully connected layer is actually the hidden layer part of the multi-layer perceptron. “Full connection” means that each neuron in the upper layer and the next layer is connected to each other, and there is no connection between the neuron nodes in the same layer. The output of the convolutional layer and the pooled layer represents the advanced features of the input image. The purpose of the fully connected layer is to use these features for classification.

![Diagram of the full connected layer](image)

Figure 3. Architecture of the full connected layer

After the image features of the corresponding channels are extracted, the features need to be merged and fused to obtain joint features. The merge process is as follows:

\[
H = [H_1, H_2, ..., H_p]^T
\]  

(5)

Where: \( p \) is the total number of channels, \( H \) is the parameter matrix of the current channel feature image, and \( T \) is the transpose. Assuming that a single channel feature outputs a three-dimensional matrix of size \( m \times m \times n \), the combined features are three-dimensional matrices of \( m \times m \times (p \times n) \), and the number of features is huge.

In order to simplify the calculation of the classifier and to efficiently extract high-dimensional feature vectors, a spatial pyramid pooling algorithm [6] is used to process the joint features to extract abstract spatial translation-invariant features. Taking the input joint features as the base layer, the nth layer divides the data evenly into \( n \times n \) sub-regions. Assume that each region \( c \) includes \( a \) points, and the feature dimension of each point is \( b = p \times n \), then this block is an \( a \times b \) dimensional matrix. For each row of the feature, take the maximum value as the final feature value. Then block \( c \) is finally reduced to a \( b \) dimensional vector:

\[
c_i = \max (c_{i1}, ..., c_{ia})
\]

(6)

\[
F(c) = \{c_1, ..., c_b\}
\]

(7)

Where: \( i \) is the number of the block, \( a \) is the number of the midpoint of the block, and \( F(c) \) is the \( b \)-dimensional vector finally obtained. For the same output data, different layer values are selected. Assuming that the picture is divided into \( r \) blocks, the final feature result is a two-dimensional matrix of \( r \times b \) dimensions. Therefore, reducing the feature matrix from three dimensions to two dimensions reduces the amount of calculation.

3. **Experiment result and analysis**

3.1. **Experimental environment and data**

In order to verify the effectiveness of the proposed algorithm, the proposed algorithm was tested on a PC with a CPU frequency of 2.5GHz, a memory of 4GB, an operating system of Windows 10, a simulation software of Python 3.6, and a deep learning framework of Tensorflow. The experimental
data include eight types, including IC chip, capacitor, resistor, inductance, diode, LED, speaker and transistor. In order to solve the lack of data samples, we rotate the pictures in each type of sample data by 90°, 180°, and 270° and put them in the database. Each category has 200 photos, of which 4500 photos are used for training and 300 photos are used for testing. After the data set is created, it is put into the designed network model.

3.2. Result analysis

In order to evaluate the performance of the component image classification method proposed in this paper, it is compared with two existing deep learning-based image classification methods. The first comparison algorithm is based on the classification of electronic components in convolutional neural networks [7]. This method performs a simple preprocessing of the image and uses the AlexNet network model to classify the images of electronic components. The second comparison algorithm proposes a classification based on high-level visualization features and migration learning deep network for target objects in the sky [8].

The classification confusion matrix of the three algorithms is calculated, and the result is shown in Figure 3. The horizontal axis of the confusion matrix represents the predicted label value of each type of component picture, the vertical axis represents the true label value of each type of component picture. The darker diagonal value represents the classification accuracy of each type of component picture. Actually, the darker the color, the higher the accuracy.

It can be seen from Figure 4 that the comparison algorithm Alexnet has some serious errors in the individual categories, and the comparison algorithm SAE-CNN has improved the classification error, and the proposed algorithm NH-CNN has the highest classification accuracy.

Typically, algorithm performance is verified by Precision, Recall, Accuracy, and $F_1$. Precision represents a ratio that is truly positive in a sample that is predicted to be positive. Recall indicates the ratio of the true positive case (TP) predicted as a positive example to all true positive cases. Accuracy
represents the ratio of predicted pairs in all samples. The $F_1$ value represents the harmonic mean of the accuracy rate and the recall rate. Table 1 shows the performance parameters of the above four algorithms.

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

(8)

$$\text{Recall} = \frac{TP}{TP+FN}$$

(9)

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

(10)

$$F_1 = \frac{2TP}{2TP+FP+FN}$$

(11)

| Algorithm     | Performance | AlexNet | SAE-CNN | NH-CNN |
|---------------|-------------|---------|---------|--------|
| Precision     |             | 0.8705  | 0.9333  | 0.9418 |
| Recall        |             | 0.8549  | 0.8933  | 0.9304 |
| Accuracy      |             | 0.8634  | 0.9158  | 0.9426 |
| $F_1$         |             | 0.8627  | 0.9133  | 0.9417 |

4. Conclusion

Aiming at the defects of traditional electronic component image recognition and classification methods that cannot effectively combine multiple deep learning features and poor classifier performance, this paper proposes an electronic component recognition and classification method based on a novel hierarchical convolutional neural network (NH-CNN) model. Compared with the existing electronic component image recognition and classification method based on convolutional neural network model, the results show that the proposed method has the highest recognition and classification accuracy. Due to the hierarchical convolutional neural network model proposed in this paper applies the feature maps obtained by convolutional autoencoder to the convolutional neural network, which can greatly improve training efficiency. Furthermore, the feature fusion method is used to output the obtained features to the fully connected layer, which is used to better express the depth information contained in the electronic component images and improve the accuracy of classification. Experiments show that the proposed algorithm has better image classification ability.

References

[1] Huang, G., Shu, J., Zhu, X., et al. (2019) Robot vision recognition and sorting strategy based on transfer learning. Computer Engineering and Applications., 8: 232-237.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation.

[3] Girshick, R., (2015) Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision., 1440-1448.

[4] Ren, S., He, K., Girshick, R., et al. (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems., 91-99.

[5] Ding Y., Li Y., Li B., (2016) Gamma correction method based on features of different brightness regions of images. Computer Technology and Development., 6: 37-39.

[6] He K., Zhang X., Ren S., et al. (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence., 9: 1904-1916.

[7] Chen X., Yu J., Chen X., et al. (2018) Classification of Electronic Components Based on Convolutional Neural Networks. Wireless Communication Technology., 2:7-12.

[8] Chen Y., Meng H., et al. (2018) Classification methods of a small sample target object in the sky based on the higher layer visualizing feature and transfer learning deep networks, EURASIP Journal on Wireless Communications and Networking 2018:127.