Scientific Research Publishing

Advances in Molecular Imaging, 2018, 8, 69-87
http://www.scirp.org/journal/ami
ISSN Online: 2161-6752
ISSN Print: 2161-6728

Fine-Grained Classification of Product Images Based on Convolutional Neural Networks

Tongtong Liu, Rubing Wang, Jikang Chen, Shengliang Han, Jimin Yang*

School of Physics and Electronics, Shandong Normal University, Jinan, China
Email: *jmyang@sdnu.edu.cn

How to cite this paper: Liu, T.T., Wang, R.B., Chen, J.K., Han, S.L. and Yang, J.M. (2018) Fine-Grained Classification of Product Images Based on Convolutional Neural Networks. *Advances in Molecular Imaging*, **8**, 69-87.
https://doi.org/10.4236/ami.2018.84007

Received: October 8, 2018
Accepted: October 23, 2018
Published: October 26, 2018

Abstract

With the rapid development of the Internet of things and e-commerce, feature-based image retrieval and classification have become a serious challenge for shoppers searching websites for relevant product information. The last decade has witnessed great interest in research on content-based feature extraction techniques. Moreover, semantic attributes cannot fully express the rich image information. This paper designs and trains a deep convolutional neural network that the convolution kernel size and the order of network connection are based on the high efficiency of the filter capacity and coverage. To solve the problem of long training time and high resource share of deep convolutional neural network, this paper designed a shallow convolutional neural network to achieve the similar classification accuracy. The deep and shallow convolutional neural networks have data pre-processing, feature extraction and softmax classification. To evaluate the classification performance of the network, experiments were conducted using a public database Caltech256 and a homemade product image database containing 15 species of garment and 5 species of shoes on a total of 20,000 color images from shopping websites. Compared with the classification accuracy of combining content-based feature extraction techniques with traditional support vector machine techniques from 76.3% to 86.2%, the deep convolutional neural network obtains an impressive state-of-the-art classification accuracy of 92.1%, and the shallow convolutional neural network reached a classification accuracy of 90.6%. Moreover, the proposed convolutional neural networks can be integrated and implemented in other colour image database.

Keywords

Product Classification, Feature Extraction, Convolutional Neural Network (CNN), Softmax

1. Introduction

With the popularity of the Internet and varieties of terminal equipment, online shopping has become a regular part of people's lives with the onset of websites such as Amazon, Dangdang, Taobao, and Jingdong. Customers view a large number of product images, and there is an urgent need for efficient product image classification methods. At present, most studies have mainly focused on keyword-based, label-based, and content-based image retrieval. Zhou [1] used a querying and relevance feedback scheme based on keywords and low-level visual content, incorporating keyword similarities. He [2] proposed a method based on the Multi-Modal Semantic Association Rule (MMSAR) to automatically combine keywords with visual features automatically for image retrieval. Xu [3] used Bayes with expectation maximization to learn an initial query concept based on the labeled and formerly unlabeled images, and the active learning algorithm selects the most useful images in the database to query the user for labeling. However, the keywords and labeled information can only explain the basic information of the goods, such as the name of the product name, the origin, the size, and price and so on, which are difficult to reflect the complete characteristics of the products. At last, images have more information and intuitive expression. If we set an image classification filter on a shopping website, it will be convenient for users to browse and quickly find their favorite products.

The last decade has also witnessed great interest in research on content-based image classification. Image classification based on the content is based on the image features, including image shape, color, and texture.

Jia [4] adopted a gist descriptor and three complementary features, including Pyramid Histogram of Orientated Gradients (PHOG), Pyramid Histogram of Words (PHOW), and Local Binary Pattern (LBP) to extract and describe the features of product images. Valuable product information (such as long skirts versus skirts, and turtleneck versus round collars) can be labeled based on the image features and classification algorithms. Furthermore, they combine discriminative features for the SVM classifier. Experimental results showed that the performance of the product image database (PI 100) improved significantly using features fusion.

Nilsback and Zisserman [5] used the features of the Histogram of Gradient Orientations (HOG), HSV value, and Scale Invariant Feature Transform (SIFT) combining an SVM classifier and multiple-kernel learning framework to classify flower images. The classification accuracy ranged from 76.3% to 95.2%.

Yao and Khosla [6] proposed a random forest, in which every tree node is a discriminative classifier that can combine node information and all upstream nodes. This method identified meaningful visual information of both subordinate categorization and the activity recognition database.

For fine-grained classification, Yao [7] presented a codebook-free and annotation-free approach for fine-grained image categorization of birds. Experimental results showed that the method was better than state-of-the-art classification

approaches on the Caltech-UCSD Birds database. Krause and Stark completed the fine-grained classification of 3D cars [8].

Dyrmann and Karstoft [9] presented a method that recognized a large number of plant species in color images, by designing and training a deep convolution neural network. The network achieved a classification accuracy of 86.2% for a total of 10,413 images containing 22 species.

However, it was difficult to contain features like the above-mentioned shape, color, and texture that could be applied to all product image classifications. Compared to these classification methods, Convolutional neural networks (CNNs) are one of the deep learning algorithms with strong ability to acquire features, simple structure, and few parameters [10]. Nevertheless the fine-grained classification of a category in product images is rarely observed.

In recent years, CNNs received much attention on the computer vision research community, mainly because they have proven to be capable of effectively classifying images and outperforming previous records in image recognition challenges. Most noticeably is the task by Krizhevsky, Sutskever, and Hinton [11], who in 2012 had a margin of 10.9% compared to the second-best entry in the ImageNet Large Scale Visual Recognition Challenge [12]. The ImageNet challenge distinguishes objects such as cat, car, tree, and house from 1000 different categories. CNNs are a deep learning application to images, and they stimulate the neuron's activity in the neocortex, where most thinking happens, as Lecun describes [13]. The main benefit of using CNNs is that they are traditional, fully connected neural networks and can reduce the amount of parameters to be learned. Convolution layers effectively extract high-level features with small-sized kernels and feed the features to fully connected layers. According to Rumelhart, Hinton, and Williams [14], the training of CNNs is performed through back-propagation and stochastic gradient descent.

This study proposed a novel deep CNN that has data augmentation pre-processing, feature extraction, and softmax classification. To solve the problem of long training time and high resource share of deep convolutional neural network, this paper designed a shallow convolutional neural network to achieve the similar classification accuracy. To evaluate the classification performance of the networks, experiments were conducted using a public database Caltech256 and a homemade product image database from shopping websites.

2. Data Material

2.1. Caltech256 Database

The Caltech256 database containing 256 object categories on a total of 30607 images. This paper selected 20 object categories which were similar to product images to input the deep convolutional neural network for training. Each category of images was randomly selected 100 images for training and 50 images for testing. The size of the input image was normalized to 256 × 256 during the experiment.

2.2. Homemade Database

The data used in the numerical analysis are mainly obtained from Internet-based e-commerce databases, including T-mall, Jingdong, and Amazon. As shown in Figure 1, 20 products were selected, including garments and shoes. The garments consist of trousers, sweaters, jackets, outdoor jackets, dresses, short T-shirts, down jackets, fleeces, vests, Chinese dresses, shirts, short pants, short skirts, scarves, and socks. The shoes include skateboard shoes, basketball shoes, leather shoes, climbing shoes, and running shoes. Each product has 200 images and after using data augmentation, there were 20,000 product images in which 16,000 images were used for training purposes and 4000 images for testing purposes. The image sizes are not the same. To facilitate the experiment, all images are normalized into $256 \times 256 = 65{,}536$ pixels.

Figure 1. Garments and shoes database.

3. Methods

This section describes the pre-processing of the product images and the architecture of the deep convolutional neural network is used for classification of the garments and shoes.

3.1. Data Augmentation

A CNN is translation invariant but not rotation invariant. The number of product images, however, on the website is limited and therefore, we can generate the training and testing data by rotating the original data using affine transformation. The data was thereby increased five-fold by mirroring the images horizontally and vertically and rotating them in 90˚ and 180˚ increments. After using data augmentation, there were 20,000 product images in which 16,000 images were used for training and 4000 images for testing.

3.2. Model Architecture

Several pre-trained networks for image classification exist such as AlexNet [11], VGGNet [15] and GoogleNet [16], which won the championship in the ImageNet Image Recognition Competition in different years. In 2012, the Hinton Task Force participated in the ImageNet Image Recognition Competition for the first time to demonstrate the potential of deep learning. It won the championship by building the CNN network AlexNet, which consisted of 8 layers and reached a 16.4% error rate. The basic composition of VGGNet is similar to AlexNet, and is also characterized by continuous convolution and large amounts of computation. VGGNet consists of 19 layers and reached a 7.3% error rate. GoogleNet, the champion model of the ImageNet competition in 2014, proved that more convolution and deeper levels can obtain a better structure. **Table 1** presents the performance results of AlexNet, VGGNet, and Google Net in the ImageNet image Recognition Competition.

However, the pre-trained network was created by the ImageNet, which is the largest database of image recognition in the world, which is different from the images in this study. Therefore, a new architecture was built to create a better classification of product images. Our CNN is sketched in **Figure 2**.

The images in the database are 256 × 256 RGB images. Matlab is used to augment data and transform the data into 227 × 227 RGB images. The network accepted 227 × 227 RGB images as input and output a vector for each block, as illustrated in **Figure 2**. The network had one 7 × 7 convolution layer with a stride

Table 1. Performance of Alex NET, VGG Net and Google Net.

	Alex Net	VGG Net	Google Net
Number of layers (layer)	8	19	22
Filter size	11,5,3	3	7,1,3,5
Top-5 Error rate (%)	16.4	7.3	6.7

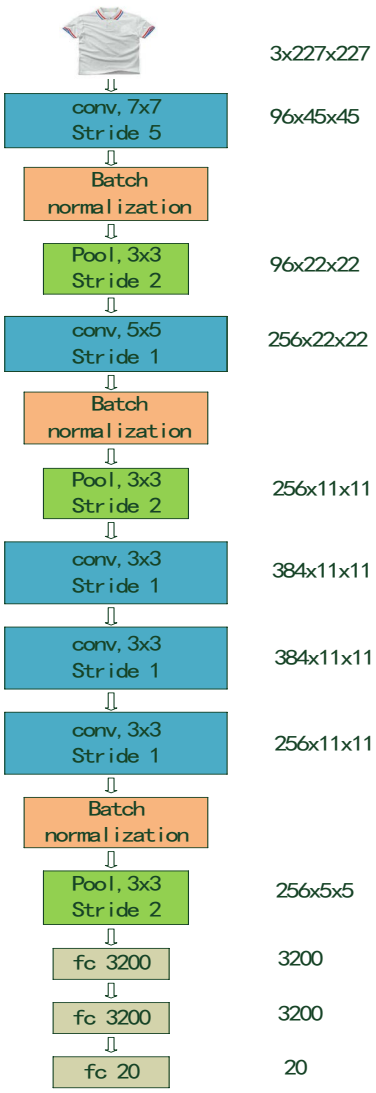

Figure 2. Deep convolutional neural network architecture.

of 5, followed by a 3 × 3 max-pooling layer with a stride of 2. This was mapped into a 5 × 5 convolution layer, which increased the number of filters from 96 to 256. Next, a 2 × 2 max-pooling layer was mapped with a stride of 2, and the number of filters was increased from 256 to 384. Following this, there were three 3 × 3 convolution layer with strides of 2. Finally, the network contained three fully connected layers, consisting of 3200 neurons and 20 softmax classifiers. In total, the network contained 628,324 learnable parameters, which is small compared to the 60 M parameters of AlexNet.

During the training, a 50% dropout was used before the three fully connected layers. The hidden layer was randomly discarded with 50% for training every epoch. This prevents all feature selectors from amplifying or reducing features all of the time, which are over fitted in the case of small samples and made poor generalizations. The dropout was used to avoid these problems in the training process. The network was trained using mini-batches with 128 images per batch,

to speed up the gradient update with a learning rate set to 0.001.

3.3. Input Layer

Data is fed to the network and the input layer produces an output vector as input to the convolution layer. Input data can be either raw image pixels or their transformations, which emphasize specific aspects of the image. This study inputs three-channel product images through a data augmentation method.

3.4. Convolution Layers

The convolution layer is the feature extraction layer. The input of each neuron is connected to the local receptive field of the previous layer, and the local feature is extracted. One of the important features of the convolution operation is that it enhances the original signal characteristics and reduces the noise. Filter kernels are slid over the original image and for each position, the dot product between the filter kernel and the part of the image covered by the kernel is determined. The calculation of the convolution layer is

$$x_j^{(l)} = f\left(\sum_{i \in M^{l-1}} x_i^{l-1} * k_{ij}^{(l)} + b_j^{(l)}\right) \tag{1}$$

where l is the number of layers, k_{ij} represents a convolution kernel with the connection of map j in the l layer and map i in the $l-1$ layer, x^{l-1} is the input feature maps of the $l-1$ layer, * represents convolution, b is the bias, and $f(\cdot)$ is the nonlinear activation function.

3.5. Max-Pooling Layers

The max-pooling layer is a method of aggregate statistics that uses the maximum or mean value of the region to reduce spatial size of a feature map and provide invariance to the network. Max-pooling layers can reduce the image size of the next layer, thereby reducing the parameters and calculations of the network. This is done by only keeping the maximum value within a $k \times k$ neighborhood in the feature map.

3.6. Batch Normalization

The role of batch normalization [17] is to normalize input data in the same range, even though the earlier layers were updated. According to Dieleman S, during each stochastic gradient descent (SGD), the corresponding activation was normalized by the mini-batch, so that the mean value of the result (output signal in each dimension) was 0 and the variance was 1 [18]. The calculation of the batch normalization is

$$y = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} x + \left(\beta - \frac{\gamma . \mu}{\sqrt{\sigma^2 + \varepsilon}}\right) \tag{2}$$

where μ and σ are the mean value and variance of the image batch x, and γ and β are trainable parameters that are updated after each batch. ε is a small constant

value that is added to the variance to avoid division by zero.

3.7. Activation Functions

The activation functions in deep learning are responsible for applying a non-linear function to the output of the previous layer. Sigmoid, tanhyperbolic (tanh), rectified linear unit (ReLU) and softplus are commonly used in deep learning.

The non-linear Sigmoid function has a large signal gain in the central region, and relatively small signal gain on both sides [19]. The output of the sigmoid function is mapped into the internal of 0 and 1, so it has a good effect on the feature space map of the signal. However, this kind of activation function cannot solve the vanishing gradient problem and is slow in network training. The calculation of the sigmoid function is

$$f(x)=\frac{1}{1+e^{-x}} \tag{3}$$

The non-linear tanh function converges faster than the sigmoid function. It is mapped into the internal of −1 and 1, and the output is centered at 0. Still, the tanh function (like the sigmoid function) cannot solve the vanishing gradient problem. The calculation of the tanh function is

$$\tanh(x)=\frac{1-e^{-2x}}{1+e^{-2x}} \tag{4}$$

This study used ReLU as the activation function. In 2011, the ReLU activation function was proposed by Glorot [20]. According to Krizhevsky [11], the ReLU function effectively suppressed the vanishing gradient problem with a faster convergence rate in training gradient descent than traditional saturated nonlinear functions. They can speed up training and keep the gradient relatively constant in all network layers. The ReLU is defined as

$$f(x)=\max(0,x) \tag{5}$$

The rectifier function is one-sided and therefore does not enforce a sign symmetry or antisymmetry. However, the response to the opposite of an excitatory input pattern is 0 (no response). Therefore, it is more biologically plausible and provides good results.

A smooth approximation to the rectifier is the softplus function. The softplus is not completely one-sided, so it is less biologically plausible and is not used as widely as ReLU. The calculation of the softplus function is

$$\text{softplus}(x)=\log\left(1+e^{x}\right) \tag{6}$$

where x is the value of input signal.

Figure 3 shows the corresponding curves of the activation functions.

3.8. Fully Connected Layers

According to traditional neutral networks, all inputs in fully connected layers are

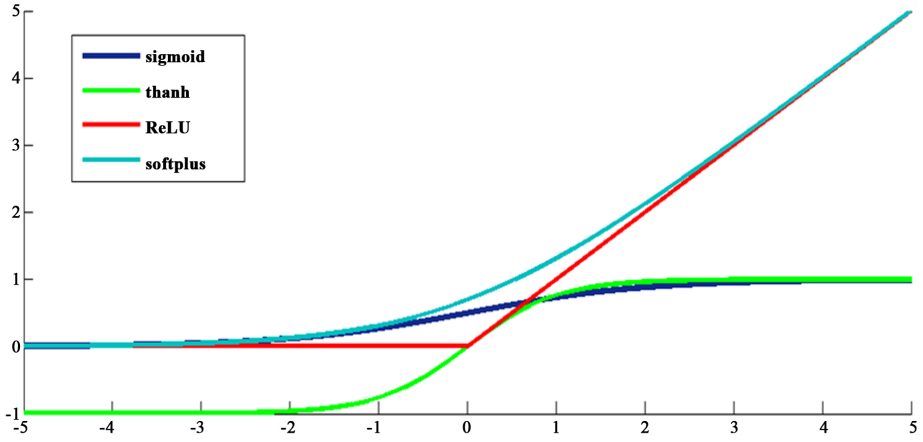

Figure 3. Activation function curves.

connected to all outputs of the previous layer. The fully connected layers are used as a way of mapping spatial features to image labels. After being trained, the network can extract features in these layers to train another classifier.

3.9. Softmax

This study used the softmax classifier, which is the generalization of the logistic model on multiple classification. The softmax classifier is an algorithm that divides the target variable into several classes. Supposing there are N input images $\{x_i, y_i\}_{i=1}^{N}$, each image is marked with k classes $y_i \in \{1,2,3,\cdots,k\}, k \geq 2$; in this study, $k = 2$. For the given test image x_i, the approximate value $p(y_i = j|x_i)$ of each class j is estimated by the hypothetical function. The calculation of the hypothetical function $h_\theta(x_i)$ is

$$h_\theta(x_i) = \begin{bmatrix} p(y_i = 1)|x_i;\theta \\ p(y_i = 2)|x_i;\theta \\ \vdots \\ p(y_i = k)|x_i;\theta \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x_i}} \begin{bmatrix} e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ \vdots \\ e^{\theta_k^T x_i} \end{bmatrix} \tag{7}$$

where $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x_i}}$ represents the normalization of the probability distribution, that is, the sum of all probabilities is 1. θ is a parameter of the softmax function. The calculation of the loss function is

$$J(x, y, \theta) = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{k} 1\{y_i = j\} \log \frac{e^{\theta_j^T x_i}}{\sum_{j=1}^{k} e^{\theta_j^T x_i}}\right] \tag{8}$$

where $1\{y_i = j\}$ is an indicative function. The rule of value is as follows: $1\{\text{the value of expression is true}\} = 1$, $1\{\text{the value of expression is false}\} = 0$. Finally, the error function is minimized by stochastic gradient descent.

3.10. Filter Capacity

In this study, the efficiency of the network was determined by evaluating the filter capacity and coverage of the network [21]. The filter capacity is a measure of the filter's ability to detect complex structures in an image. If the capacity is small, only local features in the image will be mapped to the next layer. On the contrary, if the capacity is large, the filter will find complex structures of elements that are not neighbors in the input image. The filter capacity is calculated as the ratio between the real filter size and the receptive field [22]. The calculation of the capacity is

$$\text{Capacity} = \frac{\text{real filter size}}{\text{receptive field}} \tag{9}$$

where the real filter size is the size of the kernel, which consists of downsampling (striding or pooling) of previous layers. If no downsampling is applied, the real filter size is the same as the kernel size. For example, if the input to a layer with kernel size n × n is downsampled by a factor k, the real filter size would then be kn × kn. In this network, there are two 3 × 3 max-pooling layers and a 2 × 2 max-pooling layer. After the first 3 × 3 max-pooling layer, the real filter size would be 3 n × 3 n. After the second 2 × 2 max-pooling layer, it would be 6 n × 6n and after the third 3 × 3 max-pooling layer, it would be 18 n × 18 n. The receptive field is defined as the region in the original image that a particular CNN's feature is focused on [22]. Increasing the size of filters in the convolution layers or using pooling can increase the receptive field and thus the filter capacity. According to Cao [21], the network is meaningless if the capacity is smaller than 1/6. For this network, the filter capacity is between 20.4% and 100%, and thereby well above the lower 1/6 limit.

3.11. Coverage

Coverage is a measure to "see" a part of the input image of the layer in a CNN. Adding convolution or pooling layers can increase coverage. The coverage of the network in the end should not exceed 100%. If coverage exceeds 100%, it will be a waste of network calculations, because the network can operate images larger than the input image. For this network, the convolution filters covered 55.9% of the input image and never exceeded the size of the image. **Table 2** shows the coverage and capacity of the network.

4. Results and Discussion

The operating system used in the experiments is Centos 7, and four Navida TITIAN X graphics cards are used. The framework used is caffe, and the analysis of the experimental results is all based on caffe.

The classification accuracy of deep convolutional neural network on Caltech256 database reached 94.8%. It shows the effectiveness of the proposed deep network and its suitability for feature extraction of color images.

Table 2. Coverage and Capacity of the Network.

	Coverage (%)	Capacity (%)
Conv1	3.08	100
Conv2	25.1	26.3
Conv3	42.7	27.8
Conv4	51.5	23.1
Conv5	60.4	19.7

Figure 4 shows the classification accuracy and cross entropy loss of the experiment on homemade database. To achieve the highest accuracy possible without overfitting the network, the training was set to 100 epochs. The average classification accuracy of the test was 92.1%. Setting appropriate learning rates in the experiment can improve the learning efficiency of network and therefore improve the classification accuracy. The learning rate was reduced three times before the experiment was stopped. At the beginning, we set the learning rate at 0.001. The test accuracy rapidly increased and the test loss rapidly declined. According to the decline of train loss curve, the learning rate of the network is relatively high. After 10 epochs, the test accuracy slowly increased, even decreasing, and the test loss was an upward trend. We therefore set the learning rate at 0.0005. It was observed that the test accuracy of the network increased again and the test loss slowly decreased. After 20 epochs, the test accuracy and test loss was not stable. We set the learning rate at 0.0001. It was observed that the test accuracy was high and the train loss continued to decline, then stabilized after 30 epochs.

Figure 5 shows the confusion matrix of the misclassification fraction for each of the 20 species. Here, it is seen that jackets (#2), shirts (#10), short pants (#11), short skirts (#12), and socks (#14) were often correctly classified with an accuracy of 96%, 96%, 97%, and 95%, respectively. However, there was no clear species that trousers (#0), sweaters (#1), outdoor jackets (#3), dresses (#4), short T-shirts (#5), down jackets (#6), fleeces (#7), vests (#8), and scarves (#13) was confused with. The classification for these species ranged from 90% to 95%. Chinese dresses (#9), skateboard shoes (#15), basketball shoes (#16), and leather shoes (#17) were often misclassified. Of these three species, only 89%, 85%, 82.5%, and 88.5% were classified correctly. Skateboard shoes (#15) were often classified with leather shoes (#17), and basketball shoes (#16) were often classified with skateboard shoes (#15), because they are similar in shape and texture. Leather shoes (#17) were often misclassified with climbing shoes (#18) and running shoes (#19), because they are similar in color and shape. The classification accuracies for these three species were, however, still well above random assignment.

Overall, most species had the highest classification accuracies. This is because the aim of the training was to obtain the most correctly classified product images, without taking into account how these product are distributed among the

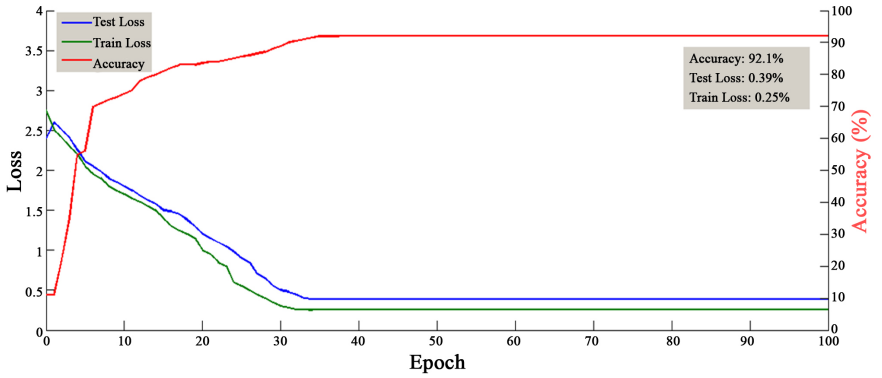

Figure 4. Classification accuracy and cross entropy loss of experiment. Red line represents the test accuracy, blue line represents test loss, green line represents train loss.

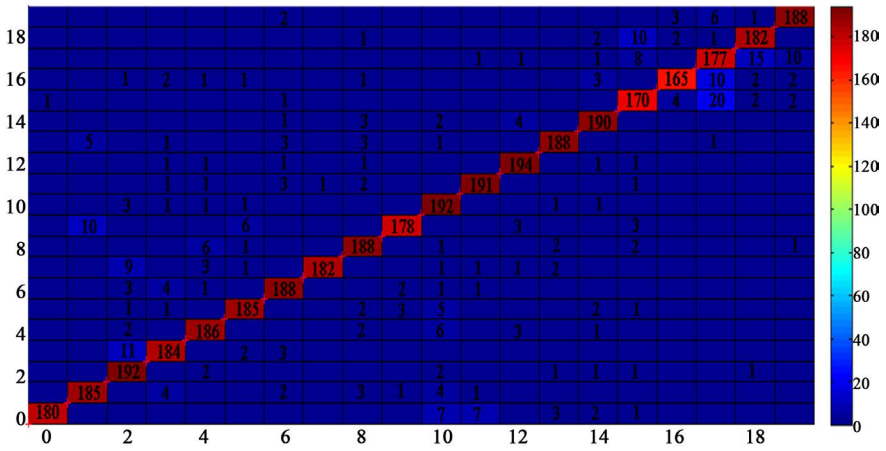

Figure 5. Confusion matrix.

20 classes. Few image samples contributed less to the overall loss. The average classification accuracy of product images was 92.1%. The classification accuracy of garments was 93.4%, and the accuracy of shoes was lower at 88.2%. This was because the shoes sample that we chose was similar and the features could not be better extracted.

As shown in **Figure 6**, we chose an image from each of the three categories, including short skirt, trouser, and basketball shoes, to show the visualization feature images of each convolution layer. It can be seen from the horizontal comparison of the feature images of each category that the first convolution layer (conv1) shows the edges, shapes, and colors of the product. Conv2 shows the texture of the product. After conv3, the feature images of product are more ambiguous and have no specific meaning. The classification accuracy of short skirts, trousers, and basketball shoes was 97%, 90%, and 82.5%. It can be seen from the vertical comparison of the feature images of each category that the edge sharpness of skirt is higher than trouser, and the trouser is higher than basketball shoes after conv3. It can be also proven from **Table 3**, which shows the mean and standard deviation value of each convolution layer feature extraction.

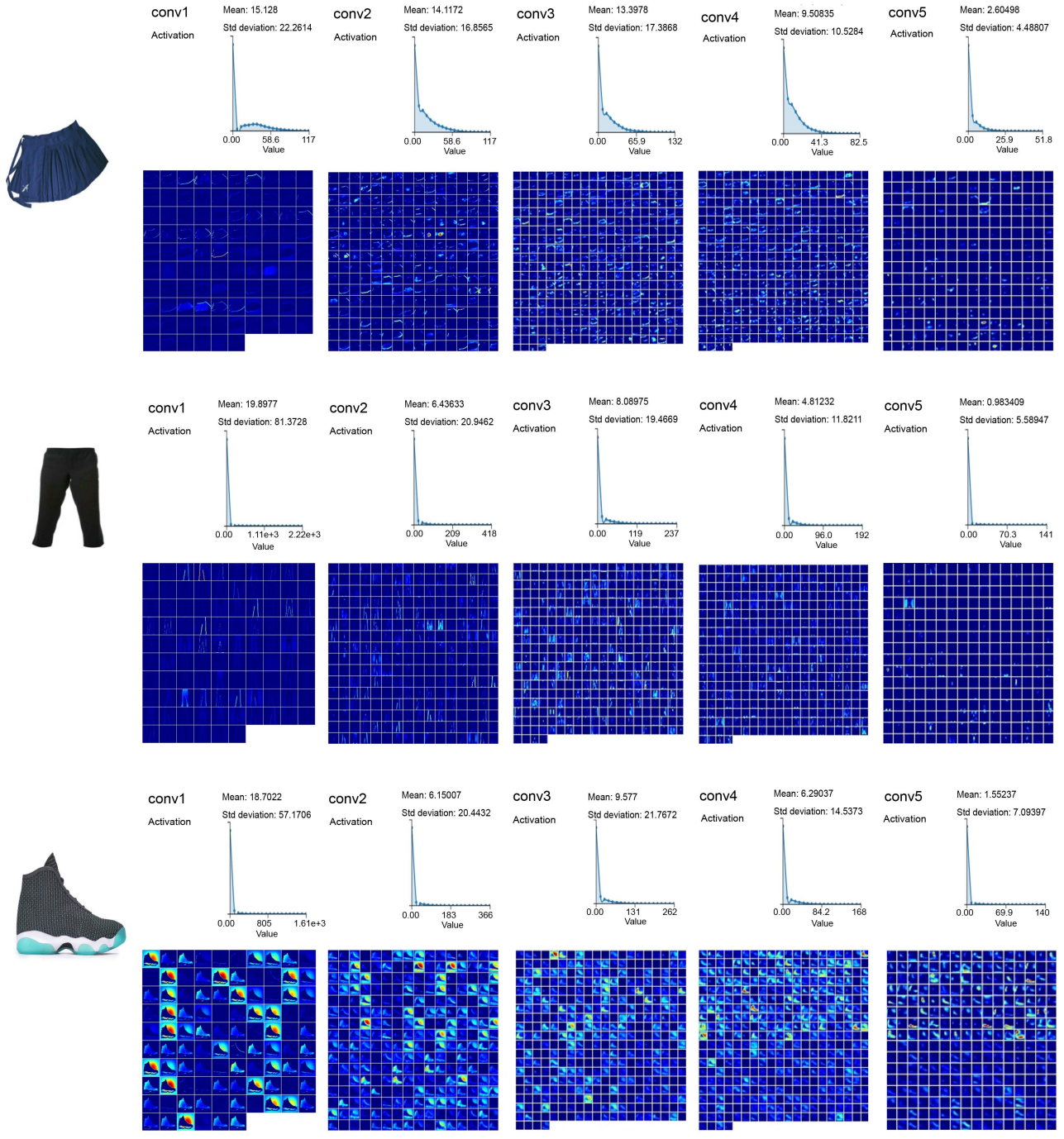

Figure 6. Visualization feature images of each convolution layer. (a) Short skirt; (b) Trouser; (c) Basketball shoes.

Table 3. The mean and standard deviation value of each convolution layer feature extraction.

		Conv1	Conv2	Conv3	Conv4	Conv5
Short skirt	Mean	15.13	14.12	13.40	9.51	2.60
	Standard deviation	22.26	16.86	17.39	10.53	4.49
Trouser	Mean	19.90	6.44	8.90	4.81	0.98

Continued

	Standard deviation	81.37	20.95	19.47	11.82	5.59
Basketball shoes	Mean	18.70	6.15	9.58	6.29	1.55
	Standard deviation	57.17	20.44	21.77	14.54	7.09

From conv1 to conv5, the mean and standard deviation value of each category are gradually decreasing. This means that the feature information of images are extracted in a stable fashion. From conv3 to conv5, the standard deviation value of short skirt is less than trouser, and trouser is less than basketball shoes. The smaller the standard deviation value, the better the effect of feature extraction and the more stable the image feature.

5. Comparative Experiment Based on Shallow Convolutional Neural Network

In the application of modern technology, saving time cost and resource share rate are very important aspects that cannot be ignored. In a relatively simple task, such as collecting fewer images in the object, the shallow convolutional neural network can accomplish the task better, why should we design a complex network with higher time cost?

5.1. Image Preprocessing

There were 4000 images in our database and each product has 200 images in which 150 images were used for training purposes and 50 images for testing purposes. The image sizes are not the same. To facilitate the experiment, all images are normalized into $256 \times 256 = 65{,}536$ pixels.

Because of the small number of samples and the shallow network layers, this paper focuses on image preprocessing. In order to eliminate the influence of complex background on the network, a more intuitive method is to extract the recognition object from the image and then use the extracted region for training. It is necessary to detect the target object in the image, and the RCNN algorithm is the classical algorithm in deep learning for detecting target object. The RCNN algorithm was proposed by Girshick [23] in 2014 and achieved great success. The detection rate on PASCALVOC database was greatly increased from 35.1% to 53.7%.

Although RCNN has achieved good results, there are some obvious shortcomings, such as the number of bounding boxes is too large, the training time is long, and many bounding boxes overlap each other, resulting in repeated calculation. To solve these problems, an improved Fast-RCNN [24] has been proposed. The biggest difference between Fast-RCNN and RCNN is that the Fast-RCNN maps all bounding regions to the last convolution layer of the network, and then uses a ROI pooling layer to unify the sizes of different bounding regions. Only one feature extraction is needed for an image, and feature extrac-

tion is not performed for each bounding region, thereby greatly improving the efficiency of calculation.

Although the speed of Fast-RCNN is greatly improved compared to RCNN, there is still a need to optimize the large number of bounding regions. In view of this, the Faster-RCNN [25] algorithm is proposed. Faster-RCNN is characterized by extracting bounding regions from feature maps after the convolution layer rather than from the original image, so a Region Proposal Networks (RPN) is added to generate bounding regions based on Faster-RCNN.

This paper used Faster-RCNN to detect the location of clothing in the image, and then the image is normalized to 64 × 64 as input image.

5.2. Fine Tune and Training

Convolution neural network as a deep learning network structure requires a lot of data for training and a deep network structure in order to achieve better classification impacts. The training result based on the small samples and shallow convolutional neural network is often unsatisfactory. In view of this situation, this paper uses ImageNet database, which consists of 1.2 million images and 1000 categories for the shallow network pre-training. Network training is a process to update the initialization parameters to the optimal parameters. When the ImageNet training is completed, the trained parameters are stored in the shallow convolutional neural network. Then, input preprocessed product database for network training to obtain optimal parameters. The network storing the optimal parameters serves as a new shallow network model for feature extract. Finally, the Softmax is used to classify these features.

5.3. Shallow Convolutional Neural Network Model Architecture

Figure 7 shows the shallow convolutional neural network architecture and the trained parameters. The database was normalized to 64 × 64 RGB images after the preprocessing. The shallow network accepted 64 × 64 RGB images as input and output a vector for each block. It had one 8 × 8 convolution layer with a stride of 1, followed by a 3 × 3 max-pooling layer with a stride of 2. This was mapped into a 6 × 6 convolution layer, which increased the number of filters from 16 to 28. Next, a 3 × 3 max-pooling layer was mapped with a stride of 2, and the number of filters was 28. Following this, there were three 4 × 4 convolution layer with strides of 1. Finally, the feature maps were mapped into a 3 × 3 max-pooling layer with a stride of 2. Then, using softmax classifiers to classify 20 category product images. The network was trained using mini-batches with 25 images per batch and the training was set to 50 epochs, to speed up the gradient update with a learning rate set to 0.001.

5.4. Results and Discussion

Figure 8 shows the classification accuracy and cross entropy loss of the experiment on homemade database. To achieve the highest accuracy possible without

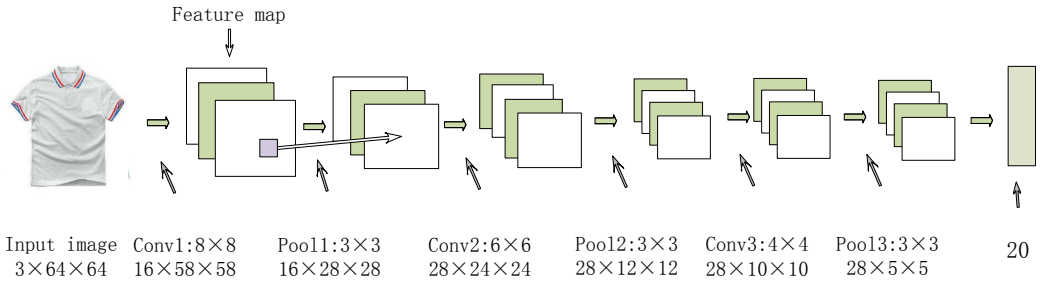

Figure 7. Shallow convolutional neural network architecture.

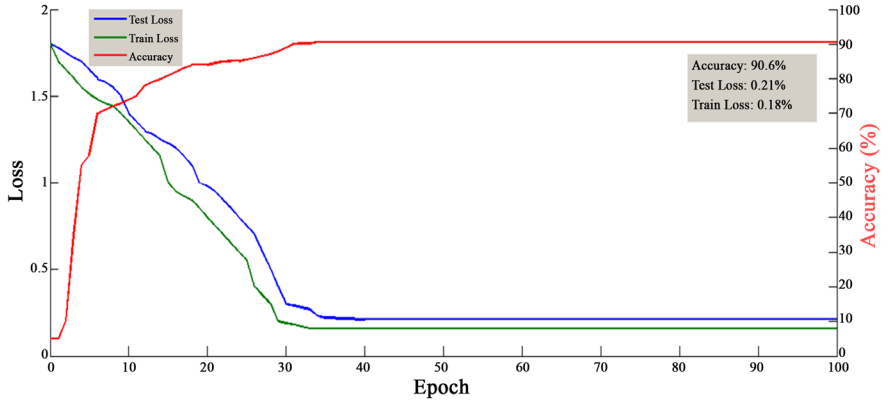

Figure 8. Classification accuracy and cross entropy loss of experiment. Red line represents the test accuracy, blue line represents test loss, green line represents train loss.

overfitting the network, the training was set to 50 epochs. The average classification accuracy of the test was 90.6%. Setting appropriate learning rates in the experiment can improve the learning efficiency of network and therefore improve the classification accuracy. The learning rate was reduced three times before the experiment was stopped. At the beginning, we set the learning rate at 0.001. The test accuracy rapidly increased and the test loss rapidly declined. According to the decline of train loss curve, the learning rate of the network is relatively high. After 10 epochs, the test accuracy slowly increased, even decreasing, and the test loss was an upward trend. We therefore set the learning rate at 0.0005. It was observed that the test accuracy of the network increased again and the test loss slowly decreased. After 30 epochs, the test accuracy and test loss was not stable. We set the learning rate at 0.0001. It was observed that the test accuracy was high and the train loss continued to decline, then stabilized after 35 epochs.

Overall, the shallow convolutional neural network can save time cost and resource share rate by reducing network layers and training epochs. However, it is impossible to achieve high classification accuracy by simply reducing the number of network layers and iterations, which requires processing in image preprocessing and network initial parameter modulation.

6. Conclusions

In this study, we designed and trained a feature-based deep CNN for color image

classification in e-commerce domains, which are comprised of data augmentation pre-processing, feature extraction, and softmax classification. The proposed network is feasible and effective by evaluating the filter capacity and coverage of the network. To evaluate the classification performance of this technique, experiments were conducted using a homemade product image database taken from shopping websites on a total of 20,000 color images, with an average accuracy of 92.1%. Empirical results for the image database have shown that the proposed feature-based deep CNN is very competitive when compared with traditional content-based image classification for all performed experiments.

To solve the problem of long training time and high resource share of deep convolutional neural network, this paper designed a shallow convolutional neural network to achieve the classification accuracy of 90.6%.

The proposed network fine-tunes the parameters and architecture based on CNNs (as reported in this study) can be readily integrated and implemented in other image recognition and classification domains.

The potential future work involves improving new and deeper network architectures for product image classification; applying the CNN on other image databases and improving the classification accuracy by transfer learning.

Data Availability

The data used in the numerical analysis are mainly obtained from Internet-based e-commerce databases, including T-mall, Jingdong, and Amazon.

Funding Statement

This work is supported by the Key Research and Development Plan in Shandong Province under grant no.2017GGX10102.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Zhou, X.S. and Huang, T.S. (2002) Unifying Keywords and Visual Contents in Image Retrieval. *Multimedia IEEE*, **9**, 23-33.

[2] He, R., Xiong, N., Yang, L.T., *et al.* (2011) Using Multi-Modal Semantic Association Rules to Fuse Keywords and Visual Features Automatically for Web Image Retrieval. *Information Fusion*, **12**, 223-230.

[3] Xu, J. and Shi, P.F. (2004) Active Learning with Labeled and Unlabeled Samples for Content-Based Image Retrieval. *Journal of Shanghai Jiaotong University*, **38**, 2068-2072.

[4] Jia, S.J., Kong, X.W., Fu, H., *et al.* (2010) Product Images Classification with Multiple Feature Combination. *Proceedings of the* 1*st International Conference on E-Business Intelligence* (*ICEBI*2010), Atlantis Press, 446-469.

[5] Nilsback, M.E. (2009) An Automatic Visual Flora-Segmentation and Classification

of Flower Images. Oxford University, Oxford.

[6] Yao, B., Khosla, A., Li, F.F., *et al.* (2011) Combining Randomization and Discrimination for Fine-Grained Image Categorization. *Computer Vision and Pattern Recognition IEEE*, Colorado Springs, 20-25 June 2011, 1577-1584.

[7] Yao, B. and Khosla, A. (2012) Codebook-Free and Annotation-Free Approach for Fine-Grained Image Categorization. *Computer Vision and Pattern Recognition IEEE*, Providence, 16-21 June 2012, 3466-3473.

[8] Krause, J., Stark, M., Jia, D., *et al.* (2014) 3D Object Representations for Fine-Grained Categorization. *International Conference on Computer Vision Workshops IEEE*, Sydney, 2-8 December 2013, 554-561.

[9] Dyrmann, M., Karstoft, H., Midtiby, H.S., *et al.* (2016) Plant Species Classification Using Deep Convolutional Neural Network. *Biosystems Engineering*, **151**, 72-80.

[10] Sun, Y., Liu, Y., Wang, G., *et al.* (2017) Deep Learning for Plant Identification in Natural Environment. *Computational Intelligence and Neuroscience*, 2017, Article ID: 7361042.

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E., *et al.* (2012) ImageNet Classification with Deep Convolutional Neural Networks. *International Conference on Neural Information Processing Systems*, Lake Tahoe, 3-6 December 2012, 1097-1105.

[12] Russakovsky, O., Deng, J., Su, H., *et al.* (2015) ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211-252.

[13] Lecun, Y., Bengio, Y., Hinton, G., *et al.* (2015) Deep Learning. *Nature*, **521**, 436.

[14] Rumelhart, D.E., Hinton, G.E., Williams, R.J., *et al.* (1986) Learning Representations by Back-Propagating Errors. *Nature*, **323**, 533-536.

[15] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition.

[16] He, K., Zhang, X., Ren, S., *et al.* (2016) Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition*, Las Vegas, 770-778.

[17] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Computer Science*, 448-456.

[18] Dieleman, S., De Fauw, J., Kavukcuoglu, K., *et al.* (2016) Exploiting Cyclic Symmetry in Convolutional Neural Networks. 1889-1898.

[19] Mount, J. (2011) The Equivalence of Logistic Regression and Maximum Entropy-models. http://www.win-vector.com/dfiles/LogisticRegressionMaxEnt.pdf

[20] Glorot, X., Bordes, A., Bengio, Y., *et al.* (2011) Deep Sparse Rectifier Neural Networks. *International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, 315-323.

[21] Cao, X. (2015) A Practical Theory for Designing Very Deep Convolutional Neural Networks Classifier Level. Technical Report.

[22] Luo, W., Li, Y., Urtasun, R., *et al.* (2016) Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, Barcelona, 5-10 December 2016, 4898-4906.

[23] Girshick, R., Donahue, J., Darrel, T., *et al.* (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587.

[24] Girshick, R. (2015) Fast-RCNN. *Proceedings of the IEEE Conference on Computer Vision*, Santiago, 7-13 December 2015, 1440-1448.

[25] Ren, S., He, K., Girshick, R., *et al.* (2015) Faster-RCNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, Montreal, 7-12 December 2015, 91-99.

Nomenclature

B: trainable scale for network layer
γ: trainable bias for network layer
μ: mean of image batch
σ: standard deviation of image batch
×: image batch
ReLU: the rectified linear unit
GIST: Global descriptor
PHOG: Pyramid Histogram of Orientated Gradients
PHOW: Pyramid Histogram of Words
LBP: Local Binary Pattern
HOG: Histogram of Gradient Orientations
SIFT: Scale Invariant Feature Transform