HK.NCCP
International Journal of Intelligent Information and Management Science
ISSN: 2307-0692, Volume 8, Issue 6, December, 2019

Research on Feature Extraction and Transfer Learning of Convolutional Neural Network to the Style of Li Brocade

Boxiong Yang[1,2*], Heng Xiao[1,2], Tao Yang[1,2], Ping Yin[1,2]
[1]School of Information & Intelligence Engineering, University of Sanya, Sanya, 572022, China
[2]Academician Workstation of Chen Guoliang, University of Sanya, Sanya, 572022, China

Abstract: Li Brocade is exquisitely made, brightly colored, and beautifully decorated, which embodies the national characteristics of the Li nationality in spinning, weaving, dyeing and embroidery. Every pattern and picture of Li Brocade has its own style. In deep learning, a convolutional neural network (CNN or Conv Net) is a class of deep neural networks that are most commonly applied to analyzing visual imagery. VGGNet is a convolutional neural network developed by researchers from the Visual Geometry Group of Oxford University and Google Deep Mind. Both content and style can be extracted as features through the VGG-19 Deep Neural Network Model. VGG-19 is used to extract the pattern style of Li Brocade, and then the features of its style transferred to other pictures according to transfer learning. The experimental results show that the styles of Li Brocade are well extracted, migrated, and preserved. If more style features of the Li Brocade pattern are needed to be dug out, the parameter value of style layers should be set as large as possible.

Keywords: Deep learning; Feature extraction; Transfer learning; CNN; Style of Li Brocade

1. Introduction

Li Brocade is produced in the Li national residential area of Hainan Island, which has become the birthplace of China's cotton textile industry. Li Brocade is known as the earliest cotton textile in China. It is a "living fossil" in the history of the nation's textile industry, spanning a history of over 3000 years [1].

Li Brocade is exquisitely made, brightly colored and beautifully decorated which embodies the national characteristics of Li nationality in spinning, weaving, dyeing and embroidery. The geometric patterns of Li Brocade are composed of straight lines, parallel lines, squares, diamonds, triangles and so on. They are expressed in clothing with abstract patterns, reflecting some characteristics of primitive thinking. It is rich in content and beautiful in color. The brocade art of the Li nationality fully demonstrates the creative ability and artistic attainment of Women of Li Nationality. Typical patterns of Li Brocade style is as shown in Figure 1.

Figure 1. Typical patterns of li brocade style

2. Transfer Learning

Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [2]. For example, the knowledge gained while learning to recognize cars could apply when trying to recognize trucks.

Transferring the style from one image onto another can be considered a problem of texture transfer. In texture transfer the goal is to synthesis a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. For texture synthesis there exist a large range of powerful non-parametric algorithms that can synthesis photorealistic natural textures by resampling the pixels of a given source texture [2].

The definition of transfer learning is given in terms of domain and task [3]. Domain D consists of: a feature space X and a marginal probability distribution P(X), where X={x1, x2,…, xn}∈X. Given a specific domain, D={X, P(X)}, a task consists of two components: a label space Y and an objective predictive function f(•)(denoted by T={Y, f(•)}), which is learned from the training data consisting of pairs, which consist of pairs {xi, yi}, where xi∈X and yi∈Y. The function f(•) can be used to predict the corresponding label, f(x), of a new instance x.

Given a source domain DS and learning task TS, as well as a target domain DT and learning task TT , transfer learning aims to help improve the learning of the target predictive function fT(•) in DT utilizing the knowledge in DS and TS , where DS = TS , or TS= TT.

3. Operator of Style Description

This paper proposes the computing method of style description operator based on the VGG-19 neural network method. A stack of non-linear filters is defined at each layer in this neural network. As the number of layers increases, the complexity of the filter also increases [4]. For example, if there are Nl linear filters in Layer 1, then we can define the style descriptor in Layer 1 as the correlation between the style features' output in Layer 1.

To obtain a representation of the style for an input image, we design a feature space to capture texture information [5]. This feature space can be built on the top of the filter responses in any layer of the CNN network. It consists of the correlations among the different filter responses, where the expectation is taken over the spatial extent of the feature maps.

The feature correlations is shown by Gram matrix as Eq. (1)

$$G^{1} \in R^{N_1 \times N_1} \tag{1}$$

Where GI is the the inner product between the feature maps in layer I.

The sum of the inner product of feature map i and feature map j in layer I is shown as Eq. (2).

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \tag{2}$$

where F_{ik}^{l} is the activation of the ith filter in layer l.

According to the feature correlations of multiple layers, we can obtain a stationary, multi-scale representation of the input image, which captures its texture information but not the global arrangement. At the same time, we can visualize the information using these style feature spaces built on different layers of the CNN network to construct an image that matches the style representation for a given input image [6-10].

4. Model Training Steps

The main training steps are described below:

1. The output of some layers in VGG represent the content and style features of the image used. For example, ['conv4_2','conv5_2'] represents content features, and ['conv1_1', 'conv2_1', 'conv3_1', 'conv4_1'] represents style features. The code example is shown as follows.

Define a list of VGg layer names and corresponding weights for calculating content loss

CONTENT_LOSS_LAYERS= [('conv4_2', 0.5), ('conv5_2', 0.5)]

List of VGg layer names and corresponding weights defining loss of computing style

STYLE_LOSS_LAYERS = [('conv1_1', 0.2), ('conv2_1', 0.2), ('conv3_1', 0.2), ('conv4_1', 0.2), ('conv5_1', 0.2)]

2. Input the content picture into the network and calculate the output value of the content picture on the specified layer of the network (e.g. ['conv4_2','conv5_2'])

3. Calculating Content Loss

Content loss is defined as the L2 norm of the difference between the feature matrix extracted from the content image at the specified level and the feature matrix extracted from the noise image at the corresponding level, which is the square of the difference between two pixels. The content loss function Li corresponding to each layer is shown in Eq. (3).

$$L_i = \frac{1}{2 * M * N} \sum_{ij} \left(X_{ij} - P_{ij} \right)^2 \tag{3}$$

Here, X is the feature matrix of the noise picture, P is the feature matrix of the content picture, M is the length * width of P, and N is the number of channels. The final content loss is the weighted sum of each layer's content loss, and then the number of layers is averaged.

4. Input Style Pictures into the Network: Calculate the output values of style pictures at network-specific layers such as ['conv1_1','conv2_1','conv3_1','conv4_1'].

5. Calculating Style Loss: Style loss can be defined as the L2 norm of the difference between Style Image and Noise Image Characteristic Matrix. The content loss function Li corresponding to each layer is shown in Eq. (4).

$$L_i = \frac{1}{4 * M^2 * N^2} \sum_{ij} \left(G_{ij} - A_{ij} \right)^2 \tag{4}$$

Here, M is the length * width of the eigenvalue matrix, N is the channel number of the eigenvalue matrix, Gij is a Gram Matrix represents the Noise Image Feature, and Aij is a GRAM Matrix represents the Style Image Feature.

The final style loss is the weighted sum of each layer's style loss, and then the number of layers is averaged.

6. The final loss function for training is the weighting of content loss and style loss:

$$L_{total} = \alpha L_{content} + \beta L_{style} \tag{5}$$

where Lcontent is the total cumulative calculating error using Eq. (3), Lstyle is total cumulative calculating error using Eq. (4), Ltotal is the final loss, and and are weighting factors with + for the content and style reconstruction. Note, when the content image needs to be highlighted in the composite image, a larger weight should be assigned to; when the style image needs to be highlighted, a larger weight should be assigned to β.

7. When the training begins, a noise image is generated based on the content picture and noise. At the same time, noise pictures are input to the network, a loss is calculated, and then noise pictures are adjusted according to the loss. Next, the adjusted pictures are input to the network, the loss is recalculated, the adjustment is made, and the calculation is done again until the specified number of iterations is reached. At this time, noise pictures have both the content and style of content pictures, and can be saved [11-15].

5. Experiments and Tests

5.1. Implementation of experimental environment

Operating System: CentOS Linux 7;
Programming Language: Python3.5;
Software Development Tool: Sublime Text 3;
GPU: NVIDIA GTX1080Ti;
Deep Learning Framework: Tensor Flow-GPU
Convolutional Network Model: VGG-19
Download URL of VGG19 Model as follows:
http://www.vlfeat.org/matconvnet/models/beta16/imagenet-vgg-verydeep-19.mat.

5.2. Experiments of style transfer

Firstly, we select one of the patterns with typical Li Brocade features as shown in Figure 2, which is the source domain of the transfer learning base on style.

Figure 2. Ordinary landscape photography

Then, a kind of various Li Brocade styles is selected as shown in Figures 3. The deep transfer learning method is then used to transfer Li Brocade's style to the picture of Figure 4. The program passes through 600 iterations, in which the results are saved after 100 times.

Figure 3. A kind of various Li Brocade styles

According to the method described above, the six iterations for the image in Figure 5 are respectively shown in Figure 4.

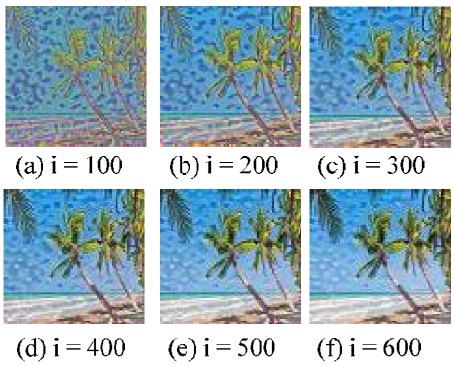

Figure 4. The results of each stage after six transformations

6. Conclusion

The following conclusions can be drawn from the above experimental results:

1. Both content features and style features can be extracted through the VGG-19 Deep Neural Network Model. We employ CONTENT_WEIGHT and STYLE_WEIGHT of weight parameters to control whether migration tends to style or content. The more layers of STYLE_LAYERS there are the more diverse style features of patterns that can be excavated. Conversely, the deeper the burial layer is in CONTENT_LAYERS, the more abstract the features are.

2. The content of the original image will be retained better and will be clearer as the number of iterations increases. However, there is lack of style if the picture is too clear. At the same time, the stylistic features of pictures will be retained less and less. Therefore, the hidden layer in CONTENT_LAYERS can be deepened appropriately, and the style migration can be enhanced while retaining the original clarity. Regardless of the design of the content, the style transfer of Li Brocade is not affected by the content.

3. If more style features of the Li Brocade pattern are needed to be dug out, the parameter value of STYLE_LAYERS should be set as large as possible. If the number of layers is too much or too deep, then the number of iterations needed will be very large to achieve better results. The shallower the hidden layer is in CONTENT_LAYERS, the more similar the expression will be for the original content in the target image.

All source code of the software can be obtained from GITHUB, where the download website is as follows: https://github.com/ceticn1/transfer-learning-of-Brocade-Style.

7. Acknowledgements

Innovation Fund for Industry, Education and Research in Universities of the Science and Technology Development Center of the Ministry of Education in China (No. 2018A01003).

References

[1] Zhou Y., et al. Research on feature extraction techniques of Hainan Li brocade pattern. Society of Photo-optical Instrumentation Engineers Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. 2017, 160–169.

[2] Robail Yasrab. ECRU: an encoder-decoder based convolution neural network(CNN) for road-scene understanding. Journal of Imaging. 2018, 4(10).

[3] Lin K., Yang H.F., Liu K.H., et al. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. Proceedings of the 5th ACMon International Conference on Multimedia Retrieval. 2015, 499-502.

[4] He K., Zhang X., Ren S., et al. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, 770-778.

[5] Liu Z., Luo P., Qiu S., et al. Deep fashion: powering robust clothes recognitionand retrieval with rich annotations. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, 1096-1104.

[6] Chen K., Hu J., He J. A framework for automatically extracting overvoltage features based on sparse autoencoder. IEEE Transactions on Smart Grid. 2018, 9(2), 594-604.

[7] Liu Shouqiang, Yu Mengjing, Li Miao, Xu Qingzhen. The research of virtual face based on deep convolutional generative adversarial networks using tensor flow. Physica A: Statistical Mechanics and its Applications. 2019, 521.

[8] Robail Y. ECRU: an encoder-decoder based convolution neural network (CNN) for road-scene understanding. Journal of Imaging. 2018, 4(10).

[9] Wang L., Qiao Y., Tang X. Action recognition with trajectory-pooled deep-convolutional descriptors. Proceedings of Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society. 2015, 4305-4314.

[10] He K., et al. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, 770-778.

[11] Chen K., et al. Framework for automatically extracting overvoltage features based on sparse autoencoder. IEEE Transactions on Smart Grid. 2018, 9(2), 594-604.

[12] Sheng L.L., Zare M.R. Biomedical compound figure detection using deep learning and fusion techniques. IET Image Processing. 2018, 12(6), 1031-1037.

[13] Yasrab R., et al. An encoder-decoder based convolution neural network (CNN) for future advanced driver assistance system (ADAS). Applied Sciences. 2017, 7(4), 312-319.

[14] Chen K., et al. Framework for automatically extracting overvoltage features based on sparse autoencoder. IEEE Transactions on Smart Grid. 2018, 9(2), 594-604.

[15] Liu S., et al. The research of virtual face based on deep convolutional generative adversarial networks using tensor flow. Physica A: Statistical Mechanics and its Applications. 2019, 5, 521-532.