

基于卷积神经网络和边缘检测的自然纹理合成算法

张定祥¹, 谭永前^{2*}

¹贵州电子信息职业技术学院电子信息工程系, 贵州 凯里 556000;

²凯里学院大数据工程学院, 贵州 凯里 556011

摘要 基于卷积神经网络(CNN)的 VGG-19(Visual Geometry Group)模型,研究了卷积神经网络对输入纹理进行卷积时,输入纹理特征图的边缘信息对生成自然纹理效果的影响。在使用卷积神经网络的 VGG 对输入图像进行卷积时,为了防止过拟合现象,采用平均池化的方式对特征图进行处理,在一定程度上保护了特征图的边缘信息,相对采用最大池化处理特征图取得了更好的生成效果。同时,提取各层特征图的边缘信息并将其叠加到特征图中,能很好地保留纹理图像的边缘结构信息。实验结果表明,改进后的方法能取得较为理想的纹理生成效果。

关键词 图像处理; 纹理合成; 卷积神经网络; 边缘信息; 池化

中图分类号 TP391.41

文献标识码 A

doi: 10.3788/LOP56.131001

Natural Texture Synthesis Algorithm Based on Convolutional Neural Network and Edge Detection

Zhang Dingxiang¹, Tan Yongqian^{2*}

¹Department of Electronic Information Engineering, Guizhou Vocational Technology College of Electronics & Information, Kaili, Guizhou 556000, China;

²School of Big Data Engineering, Kaili University, Kaili, Guizhou 556011, China

Abstract Based on the Visual Geometry Group (VGG-19) model of convolutional neural networks (CNN), influences of the edge information in an input texture feature map on the natural texture are studied when the CNN convolves the input texture. When the input image is convoluted by the VGG using the CNN, the feature map is processed in an average pooling manner to prevent overfitting, which protects the edge information of the feature map to some extent and the generation effect is better than that obtained via max-pooling processing. The edge information of each layer of feature map is extracted and superimposed on the feature map, which preserves the edge structure information of the texture image well. Experimental results demonstrate that the proposed method achieves a good texture generation effect.

Key words image processing; texture synthesis; convolutional neural network; edge information; pooling

OCIS codes 100.2960; 110.3000; 110.2970

1 引言

纹理合成是以样本纹理图像为基础,通过相关算法重新生成一个新图像的过程,新生成的图像无论在结构上还是视觉上都与样本纹理相似。早期的纹理合成一般采用两种方式。一种是以像素为单位

进行采样生成纹理图像,一次只能合成一个像素,这类算法以文献[1-3]中基于像素点的纹理合成算法为代表。基于像素点的纹理合成算法合成速度慢,效率低下,显然达不到实时合成的需求。另一种是基于样本块的纹理合成,该方法以样本纹理为源图像,对样本纹理进行块采样,通过相关算法输出生成

收稿日期: 2018-11-13; 修回日期: 2018-12-07; 录用日期: 2019-01-22

基金项目: 贵州省教育厅青年科技人才成长项目(黔教合 KY 字[2017]335)、贵州省科技合作计划(黔科合 LH 字[2017]7161 号、黔科合 LH 字[2016]7318 号)、国家自然科学基金(11464023)

* E-mail: tanyongqian1@163.com

目标图像,以文献[4-6]中算法为代表。基于块的纹理合成技术相对于基于像素点的纹理合成技术,提高了合成的速度,是目前采用较多的合成算法。

无论是基于像素点的纹理合成还是基于样本块的纹理合成,都是基于参数的合成方法,需要对合成过程进行实时监测,根据合成结果不断调整相关参数以便得到更好的合成效果。基于参数的纹理合成过程,需要进行繁琐的参数调整,在合成质量、稳定性以及不需要用户控制等方面都还有很大的改进空间。最近几年,基于深度学习的卷积神经网络(CNN)在计算机视觉等领域取得了较好的应用,比如在物体分类、图像识别、人脸识别以及图像分割等方面都取得了较好的效果。本文把基于深度学习的CNN应用到纹理合成算法中,在总结 Gatys 等^[7]算法的不足之后提出了基于CNN和边缘检测的纹理合成算法,采用VGG-16(Visual Geometry Group)^[8-9]模型进行纹理合成训练,相对于传统的基于参数的纹理合成算法,取得了更好的合成结果。

2 CNN和图像边缘检测

2015年Gatys等^[7]把CNN用于纹理合成,取得比基于参数的纹理合成算法更好的合成结果。合成方法如下:1)原始样本纹理输入到CNN模型,在该模型中计算多个层的特征响应上的Gram矩阵

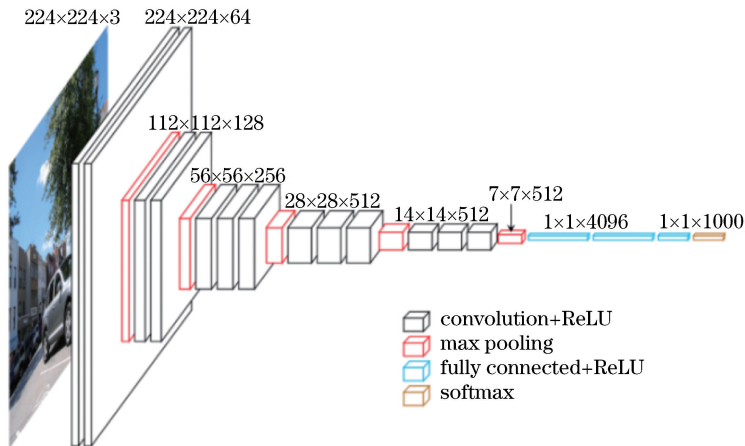


图1 VGG整体结构

Fig. 1 VGG overall structure

合成纹理模型如图2所示,其中左侧图为纹理特征图分析,源纹理样本输入到CNN,并计算大量层上的特征响应矩阵 G_l 。右侧图为纹理合成,用白噪声初始化一个随机图像 x_1 输入到CNN,计算纹理模型中每层损失函数的贡献值 E_l , L 为总损失函数^[10]。在每个像素值的总损失函数上计算梯度下降,生成与样本纹理相同的Gram矩阵,最后得到一

E_l ;2)初始化白噪声图像输入CNN模型,计算包含纹理模型在内的每个层的损失函数。在每个像素值的总损失函数上计算梯度下降,生成与原始纹理相同的Gram矩阵得到一个新图像。Gatys等提出的方法虽然能在自然纹理、随机纹理上取得比传统算法更好的效果,但在合成类似结构性纹理和半随机性纹理时会出现不稳定、亮度和对比度变化剧烈等情况。

2.1 深度卷积神经网络

VGG-19模型是基于深度学习的网络模型,该模型已应用于人脸识别、图像分类等领域。文献[7]中使用VGG-19进行纹理生成训练,VGG在加深网络层数的同时为了避免参数过多,在所有层都采用 3×3 的小卷积核,卷积层步长设置为1。VGG的输入设置为 $224 \text{ pixel} \times 224 \text{ pixel}$ 大小的RGB图像,在训练集上对所有图像计算红、绿、蓝(R、G、B)均值,然后把图像作为输入传入VGG卷积网络,使用 3×3 或者 1×1 的过滤器过滤,卷积步长固定为1。VGG全连接层有3层,激活函数整流线性单元(ReLU)用于隐层神经元输出,VGG的整体结构如图1所示。其中,convolution+ReLU表示卷积激活函数层,max pooling表示最大池化层,fully connected+ReLU表示全连接层激活函数,softmax表示最后的多分类神经网络输出和归一化层。

个新图像。

图2中 $G_{l-ij} = \sum_k F_{l-ik} F_{l-jk}$, $E_l = \frac{1}{4N_l^2 M_l^2} \times \sum_{i,j} (G_{l-ij} - \hat{G}_{l-ij})^2$, $L(x_1, x_2) = \sum_{l=0}^L w_l E_l$,其中: G_{l-ij} 表示 l 层上的特征图 i 和 j 之间的内积; F_{l-jk} 表示第 l 层位置 k 处第 j 个过滤器的激活值; k 是输入特征图

的数量; N_l 为 l 层的行数; M_l 为 l 层的列数; G_l, \hat{G}_l 分别表示 l 层上源纹理和生成纹理的 Gram 矩阵; E_l 表示第 l 层对总损失函数的贡献值; x_1 表示源纹理图像; x_2 表示生成的纹理图像; w_l 为总损失函数的权重因子。第 l 层上的 E_l 对激活函数求偏导为^[11-14]

$$\frac{\partial E_l}{\partial \hat{F}_{l,jk}} = \begin{cases} \frac{1}{N_l^2 M_l^2} [(\hat{F}_l)^T (G_l - \hat{G}_l)]_{ji}, & \text{if } \hat{F}_{l,jk} > 0 \\ 0, & \text{if } \hat{F}_{l,jk} < 0 \end{cases}, \quad (1)$$

式中: $\hat{F}_{l,jk}$ 为生成纹理第 l 层位置 k 上的第 j 个滤波器的激活值; T 表示矩阵转置。

Gatys 等提出的算法采用了深度 CNN 模型来生成自然纹理, 比传统参数化的纹理合成算法取得的合成效果更好。但 Gatys 等采用 CNN 模型来训练, 需要花费较传统算法更多的训练时间, 同时, 在卷积过程中, 采取最大池化的方式对卷积后的纹理图像进行处理, 使得纹理图像的部分信息丢失, 特别是图像的边缘结构信息, 这也是该算法只能用于生成其自然纹理的原因。

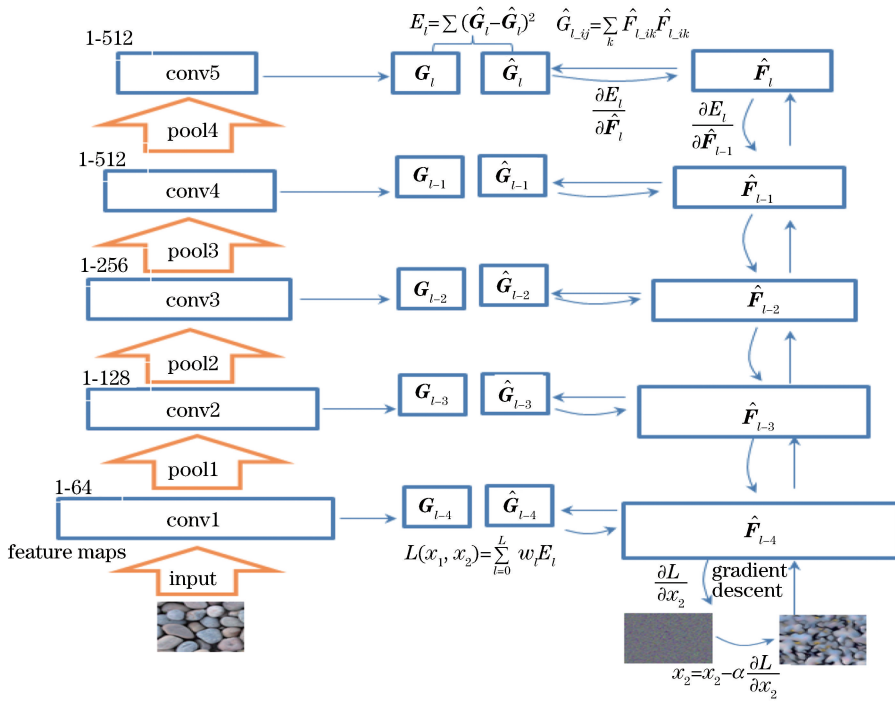


图 2 CNN 纹理合成过程

Fig. 2 CNN texture synthesis process

2.2 图像边缘检测

图像的边缘信息往往体现在图像梯度信息发生剧烈变化的区域, 图像的边缘给人们更强的视觉感受, 所以图像的边缘信息在纹理合成过程中不可忽视。Gatys 等提出的算法在对输入的纹理图像卷积后进行最大池化处理时会丢失图像的部分边缘信

息, 导致最终合成结果中边缘信息模糊不清, 影响合成纹理的视觉效果。本文对卷积过程中图像的每一层进行边缘提取, 然后把提取的边缘信息叠加到每个特征图中, 很好地保留了纹理图像的边缘结构信息。采用 Kirsch 边缘算子进行图像的边缘信息提取^[15], Kirsch 算子 8 个方向的模板分别为

$$\begin{aligned} a_0 &= \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, a_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}, a_2 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, a_3 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\ a_4 &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}, a_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}, a_6 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, a_7 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}. \end{aligned} \quad (2)$$

假设图像中任意像素点 P_A 及周围 3×3 区域的灰度如图3所示, 设 $g_i (i = 0, \dots, 7)$ 为图像经过

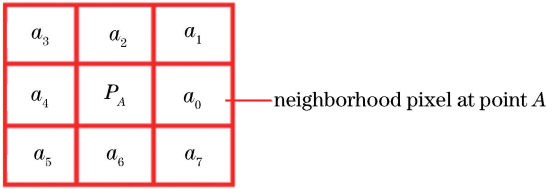


图3 任意点边缘值

Fig. 3 Edge value of arbitrary point

Kirsch 边缘算子第 $i+1$ 个模板通过卷积处理后得到的 A 点的灰度值, 则有

$$g_0 = 5 \times (a_3 + a_4 + a_5) - 3(a_2 + a_6) - 3(a_1 + a_0 + a_7), \quad (3)$$

式中: $a_i (i = 0, \dots, 7)$ 为任意点 A 的邻域像素, 使用 (3) 式同样的方法可计算 A 点其他方向的灰度值, 则处理后 A 点的灰度值为

$$g_A = \max(g_i), i = 0, \dots, 7. \quad (4)$$

使用 Kirsch 算子边缘提取效果如图4所示。

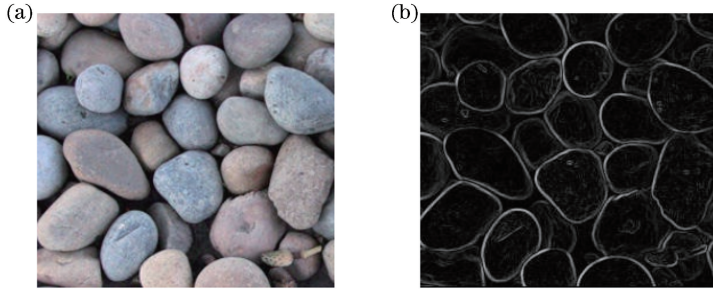


图4 Kirsch 算子边缘提取。(a)原图像;(b) Kirsch 边缘提取效果

Fig. 4 Edge extraction by using Kirsch operator. (a) Original image; (b) Kirsch edge extraction effect

3 算法步骤

改进后算法步骤如下: 1) 输入源样本纹理到 CNN 的 VGG-19 中计算其各层特征图; 2) 计算各

层特征图的边缘信息并保留; 3) 把保留的各层特征图的边缘信息进行叠加; 4) 对各层进行梯度下降, 输出 Gram 矩阵生成对应的纹理图像。改进算法流程如图5所示。

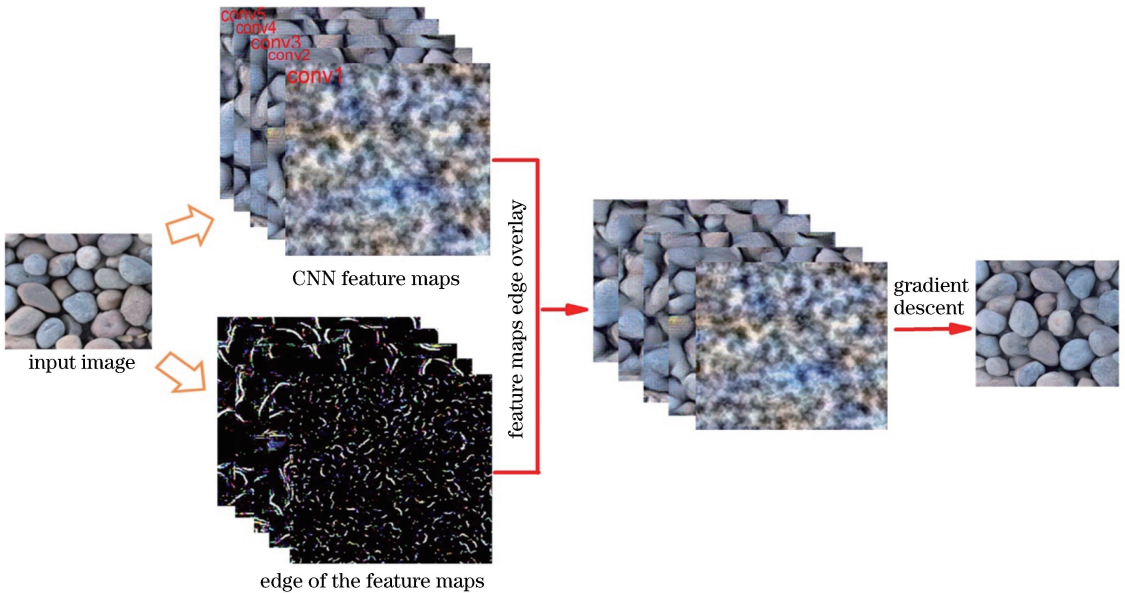


图5 改进算法流程图

Fig. 5 Flow chart of improved algorithm

4 实验结果分析

4.1 实验参数

所有实验环境为主频 3.7 GHz, 内存为 26 GB,

GPU 为 NVIDIA Titan X, Python 版本为 3.6, 网络采用 TensorFlow 框架编程实现。在实验中, 过滤器大小为 $3 \times 3 \times n$, 其中 n 是输入特征图通道, 卷积的步长和填充均为 1。输入样本图像素大小为

224×224×3 (RGB 图像), 源纹理来自数据库
www.textures.com。

4.2 实验结果对比分析

为验证算法的有效性, 对算法生成的自然纹理

与文献[4]以及文献[7]的输出效果纹理图进行了对比, 如图 6 所示。

由图 6 生成的纹理效果图对比可知, 图 6(b)是文献[4]的Image quilting算法生成的纹理图效果。



图 6 生成纹理效果图比较。(a)输入源纹理;(b)文献[4]算法效果;(c)文献[7]算法效果;(d)改进后算法效果
Fig. 6 Comparison of generated texture images. (a) Input source textures; (b) results of algorithm in Ref. [4];
(c) results of algorithm in Ref. [7]; (d) results of improved algorithm

该算法采用最小误差路径拼接样本纹理块,相对于其他基于像素的纹理合成算法,提高了纹理合成的效率。但该算法在匹配纹理相似度时,只考虑了纹理块之间的颜色误差,忽略了纹理本身的边缘结构信息,导致合成结果图中有些边缘无法很好地匹配,出现边缘信息模糊不清的情况。尤其是对边缘结构较明显的纹理图像进行合成时边缘不匹配情况更为明显,图 6(b)所示线框中为边缘信息模糊、拼接不自然、不平滑的情况。

图 6(c)为文献[7]中采用 CNN 生成的纹理图像,该方法采用基于深度学习 CNN 的 VGG-19 模型,通过对输入图像进行卷积-池化的方式计算输入图像多个层的特征图,得到每一特征图的 Gram 矩阵,再用白噪声初始化一个输入图像,用同样的方式计算得到每一特征图和相应的 Gram 矩阵,然后计算两边图像相应总的损失函数执行梯度下降,最后生成纹理图像,如图 2 所示。采用 CNN 生成纹理图像的方法打破了传统的基于参数化的纹理合成方法,第一次采用基于 CNN 模型的方式来生成纹理图像。基于 CNN 的纹理合成方法与基于样本块 Image quilting 算法生成的纹理相比,具有更加丰富的纹理内容,取得更好的视觉效果。但由于基于 CNN 的纹理生成算法为了防止过拟合,采用最大池化的方式对图像进行降维处理,各特征层会丢失图像的部分信息,尤其是图像的边缘结构信息,最终导致生成的纹理图像边缘信息不够清晰,生成纹理不稳定,如图 6(c)所示线框中出现了边缘匹配错误、边缘不自然等情况。

图 6(d)为改进后算法生成的纹理图像,该方法在使用 CNN 对输入纹理图像进行卷积后采用平均池化的方式对纹理进行处理,相对于最大池化方式,平均池化在一定程度上保护了特征图的边缘相关信息。同时改进后的算法对每一层特征图都提取边缘信息然后再叠加,最大程度地保护了纹理本身的结构信息,使得生成结果最大程度上与纹理本身结构相吻合,与文献[4]和文献[7]相比,改进后生成的纹理图像在边缘部分更加清晰、自然,取得较前两种算法更好的合成效果。但由于改进方法需对大量特征图进行边缘信息提取和叠加,纹理生成时间比文献[4]和文献[7]中的传统算法更长。对纹理生成过程进行加速优化,提高纹理生成效率,将是今后关注的重点。

5 结 论

基于深度学习 CNN,研究了纹理图像的边缘结

构信息以及基于 CNN 的 VGG-19 模型对生成自然纹理图像效果的影响。实验结果表明:1) 基于 CNN 的 VGG-19 模型在计算输入图像各层特征图时,采用最大池化的方式对各特征图进行处理,会导致各特征图丢失部分边缘信息,影响其纹理的生成效果,采用平均池化对特征图进行处理,能取得比前者更好的生成效果;2) 纹理图像的边缘结构信息对合成纹理的效果有较大影响,在计算各层特征图时保留各层特征图的纹理边缘信息,在执行梯度下降时再把各层的边缘信息与其各特征图进行叠加,最后执行梯度下降后生成的纹理能取得较好的效果;3) 基于块采样的 Image quilting 纹理合成算法采用在样本空间搜索匹配块的方式进行合成,该方法受限于有限的样本空间,导致合成的纹理图像有太多的重复纹理块,影响自然纹理的视觉效果;4) 基于 CNN 模型的纹理生成方法避免了基于块采样的纹理合成样本数量有限导致合成结果有重复块的情况,能取得较传统基于参数化的纹理合成算法更好的效果。

参 考 文 献

- [1] Wei L Y, Levoy M. Texture synthesis over arbitrary manifold surfaces[C] // The 28th Annual Conference on Computer Graphics and Interactive Techniques, August 12-17, 2001, Los Angeles. New York: ACM, 2001: 355-360.
- [2] Tong X, Zhang J D, Liu L G, *et al.* Synthesis of bidirectional texture functions on arbitrary surfaces [C] // The 29th Annual Conference on Computer Graphics and Interactive Techniques, July 23-26, 2002, San Antonio, Texas. New York: ACM, 2002: 665-672.
- [3] Ashikhmin M. Synthesizing natural textures [C] // Proceedings of the 2001 Symposium on Interactive 3D Graphics, March 19-21, 2001, North Carolina. New York: ACM, 2001: 217-226.
- [4] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer [C] // The 28th Annual Conference on Computer Graphics and Interactive Techniques, August 12-17, 2001, Los Angeles. New York: ACM, 2001: 341-346.
- [5] Xue F, Cheng C, Jiang J L. Wang Tile-based improved texture synthesis [J]. Journal of Computer Applications, 2010, 30(8): 2098-2100, 2156.
薛峰, 成诚, 江巨浪. 基于 Wang Tile 的改进纹理合成算法 [J]. 计算机应用, 2010, 30(8): 2098-2100,

- 2156.
- [6] Kwatra V, Schoed L A, Essa I, *et al.* Graphcut textures image and video synthesis using graph cuts [C] // The 30th Annual Conference on Computer Graphics and Interactive Techniques, July 27-31, 2003, San Diego, California. New York: ACM, 2003: 277-286.
- [7] Gatys L A, Ecker A S, Bethge M. Texture synthesis using convolutional neural networks [EB/OL]. (2015-12-06)[2018-11-01]. <https://arxiv.org/abs/1505.07376>.
- [8] Cadieu C F, Hong H, Yamins D L K, *et al.* Deep neural networks rival the representation of primate IT cortex for core visual object recognition [J]. PLoS Computational Biology, 2014, 10(12): e1003963.
- [9] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2015-04-10)[2018-11-01]. <https://arxiv.org/abs/1409.1556>.
- [10] Portilla J, Simoncelli E P. A parametric texture model based on joint statistics of complex wavelet coefficients [J]. International Journal of Computer Vision, 2000, 40(1): 49-70.
- [11] LeCun Y A, Bottou L, Orr G B, *et al.* Efficient backprop[M] // Montavon G, Orr G B, Müller K R. Neural networks: tricks of the trade. Heidelberg: Springer, 2012, 7700: 9-48.
- [12] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions[C] // Proceedings of the British Machine Vision Conference 2014, September 1-5, 2014, Nottingham, UK. Durham, England, UK: BMVA Press, 2014.
- [13] Jia Y Q, Shelhamer E, Donahue J, *et al.* Caffe: convolutional architecture for fast feature embedding [C] // The 22nd ACM International Conference on Multimedia, November 3-7, 2014, Orlando, Florida. New York: ACM, 2014: 675-678.
- [14] Okazawa G, Tajima S, Komatsu H. Image statistics underlying natural texture selectivity of neurons in macaque V4 [J]. Proceedings of the National Academy of Sciences, 2015, 112(4): E351-E360.
- [15] Xie J, Lu Y, Gao R, *et al.* Cooperative training of descriptor and generator networks[EB/OL]. (2018-10-29)[2018-11-01]. <https://arxiv.org/abs/1609.09408>.