

多个邻域 CNN 融合粒子群算法的边缘检测*

王红红[†], 黄永东, 江巧永

(北方民族大学 信息与计算科学学院, 银川 750021)

摘要: 受 Bastürk 算法思想的启发, 考虑多个邻域元素的影响, 并采用完全稳定的细胞神经网络克隆模板设计方法, 把 1 邻域扩展到 2 邻域和 3 邻域, 将粒子群优化算法应用于模板学习。仿真实验表明, 本算法不仅比经典的边缘检测算法更具优越性, 而且比基于遗传算法和克隆选择算法的边缘检测效果更为有效, 具有良好的应用前景。

关键词: 细胞神经网络; 克隆模板; 粒子群; 多邻域; 边缘检测

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2012)09-3501-05

doi:10.3969/j.issn.1001-3695.2012.09.080

Employing multiple neighbor of cellular neural network with particle swarm optimization algorithm for edge detection

WANG Hong-hong[†], HUANG Yong-dong, JIANG Qiao-yong

(Institute of Information & System Science, Beifang University of Nationalities, Yinchuan 750021, China)

Abstract: Inspired by Bastürk's thoughts, this paper considered the interaction among multiple cells, and adopted the method of completely stable CNN cloning template design. Then it extended one neighbor to two and three neighbors, and also used particle swarm optimization (PSO) algorithm in template learning. Simulation results indicate that the proposed method is more superior than the classic edge detectors and more effectiveness than the edge detectors based on genetic algorithm and cloning selection algorithm, and has a good application prospect.

Key words: cellular neural network; cloning template; particle swarm optimization; multiple neighbor; edge detection

在图像处理过程中,边缘检测是十分重要的,因为图像的边缘往往携带着大部分的信息。边缘通常存在于图像的不规则结构和不平稳现象中,也即存在于信号的奇异点或突变点处,这些点给出了图像的轮廓,而这些轮廓又常常是在图像处理所需的一些非常重要的特征条件,这就需要在一幅图像检测和提取出它的边缘,因而研究有效的边缘检测算法具有重要意义。经典的边缘检测的算法有 Roberts, Sobel, Prewitt, Laplacian, LOG, Canny 算子等,这些算子检测不全,且 Canny 算子在检测直线方面易发生弯曲。2003 年, Zhao 等人^[2]提出了借助 CNN 对含噪图像进行边缘检测的算法,仿真实验说明该方法具有一定的实用性,但是算法只涉及到 1 邻域的元素对细胞神经网络模板的影响,而实际中应考虑多个邻域细胞间的影响。2009 年, Bastürk 等人^[3]提出了利用差分进化算法对 CNN 模板进行优化的边缘检测算法,取得了较好的效果。但是该算法存在一些不足之处:a)只考虑了 1 邻域细胞间的影响,而没有考虑多个邻域的元素对模板的影响,如 2 邻域和 3 邻域;b)利用经典的差分进化算法对模板进行优化处理,所用种群数量过多,收敛速度慢。与粒子群算法相比,其他的优化算法也具有某些不足之处,如遗传算法^[4]稳定性不高,克隆选择算法^[5]所用种群数量多。为了克服以上不足之处,本文提出了多邻域细

胞神经网络融合粒子群算法的边缘检测,主要特点有:a)将 1 邻域扩展到 2 邻域和 3 邻域;b)本算法是完全稳定的;c)所用种群数量不多。

1 细胞神经网络

CNN^[6,7]首次是在 1988 年提出的,已经涉及很多应用领域。这些应用大部分与图像处理有关,这主要得益于 CNN 的高并行性和易于硬件实现性。

定义 1 一个二维 CNN 是一个局部互连的仿真处理数组。一个 $M \times N$ 的 CNN,有 $M \times N$ 个细胞排列在 M 行 N 列,即 i 行 j 列的细胞为 $C(i, j)$ 。

定义 2 $C(i, j)$ 的 r -邻域定义为

$$N_r(i, j) = \{C(k, l) : \max(|k-i|, |l-j|) \leq r, 1 \leq k \leq M, 1 \leq l \leq N\} \quad (1)$$

其中 r 为正整数。

由定义 2 可以得出细胞神经网络的响应范围公式为 $(2r + 1) \times (2r + 1)$, r 为邻域大小。若 $r=1$ 则为 3×3 响应范围; $r=2$ 则为 5×5 响应范围; $r=3$ 则为 7×7 响应范围。 $C(i, j)$ 电路方程如下所示。

状态方程:

收稿日期: 2012-03-06; **修回日期:** 2012-04-14 **基金项目:** 国家自然科学基金资助项目(10961001, 61102008); 国家教育部科学基金资助项目(10961001); 宁夏高校学校科研资助项目(2010JY003); 北方民族大学自主科研基金资助项目(2011ZYC033)

作者简介: 王红红(1986-),女(通信作者),山东安丘人,硕士研究生,主要研究方向为模式识别、图像处理(wanghonghong8785@163.com); 黄永东(1974-),男,教授,博士,主要研究方向为模式识别与图像处理、小波分析与信息处理; 江巧永(1985-),男,硕士研究生,主要研究方向为智能计算与智能信息处理。

$$\dot{x}_{ij} = z_{ij} - x_{ij} + \sum_{C(k,l) \in N_2(i,j)} a_{kl} y_{kl} + \sum_{C(k,l) \in N_2(i,j)} b_{kl} u_{kl} \quad (2)$$

$1 \leq i \leq M, 1 \leq j \leq M$

输出方程:

$$y_{ij} = \frac{1}{2}(|x_{ij} + 1| + |x_{ij} - 1|) \quad (3)$$

$1 \leq i \leq M, 1 \leq j \leq M$

输入方程:

$$u_{ij} = E_{ij} \quad 1 \leq i \leq M, 1 \leq j \leq M \quad (4)$$

抑制方程:

$$|x_{ij}(0)| \leq 1, 1 \leq i \leq M, 1 \leq j \leq M \quad (5)$$

$$|u_{ij}| \leq 1, 1 \leq i \leq M, 1 \leq j \leq M \quad (6)$$

参数假设:

$$A(i,j;k,l) = A(k,l;i,j), B(i,j;k,l) = B(k,l;i,j) \quad (7)$$

$1 \leq i \leq M, 1 \leq j \leq M$

其中:输入、阈值、状态与输出分别记为 u_{ij} 、 z_{ij} 、 x_{ij} 与 y_{ij} 。每个细胞有前馈突触 $b_{kl}u_{kl}$ 和反馈突触 $a_{kl}y_{kl}$, 分别为相邻细胞的输入和输出。其工作原理如图 1 所示。

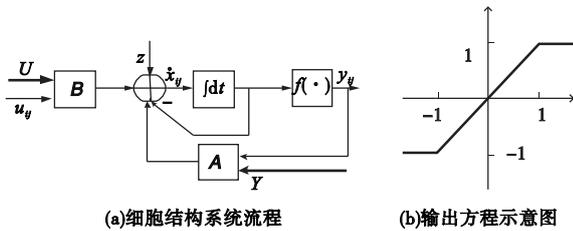


图1 单细胞工作原理

因此, CNN 处理图像的性能可以由矩阵 A 、 B 还有阈值 z 决定。CNN 是通过半径为 r 的响应范围 $N_r(i,j)$ 局部互连作用到它的相邻细胞。例如半径为 $r=1$ 的 3×3 响应范围, 其中细胞 $C_{i,j}$ 耦合到它的第一级邻域的细胞 $C_{k,l}$, 满足:

$$(k,l) = (i+1,j+1), (i+1,j), (i+1,j-1), (i,j+1), (i,j-1), (i-1,j+1), (i-1,j), (i-1,j-1) \quad (8)$$

同样可以得到 $C_{i,j}$ 在 5×5 、 7×7 响应范围的所有邻域细胞的位置, 在此就不再赘述。

对 3×3 响应范围的 CNN 而言, 完全可由 19 个参量决定, 即阈值 z 以及构成矩阵 A 、 B 的 18 个参量。由文献[7]可知, 一个完全稳定的 CNN 3×3 响应范围可以选择如下模板, 即

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_7 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_4 \\ a_3 & a_2 & a_1 \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_7 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_4 \\ b_3 & b_2 & b_1 \end{bmatrix} \quad (9)$$

显然, 采用完全稳定的 CNN 模板设计还可以降低参数维数, 即 3×3 响应范围的所有参数写成向量形式为 $u_1 = [a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5, z]$ 。同样可以得到 5×5 、 7×7 响应范围的参数向量分别为

$$u_2 = [a_1, \dots, a_{13}, b_1, \dots, b_{13}, z]$$

$$u_3 = [a_1, \dots, a_{25}, b_1, \dots, b_{25}, z]$$

CNN 模板设计方法中有三种方法^[11]: 直觉法 (intuitive way)、模板学习 (template learning) 和直接模板推导 (direct template derivation)。模板设计研究中, 设计者直觉的想法看似是

最快的方式, 但大多数时候并不能保证找到所需的模板, 主要缺点是在模板学习和数组动态中要做大量实验; 而直接模板推导法似乎取决于特定的模板类, 需要一些计算能力的模板学习需求; 应用最为广泛的模板研究方法是模板学习方法, 它需要找出输入和期望的输出对之间的相关性。本文采用模板学习方法。

2 粒子群算法

粒子群优化算法^[8,9]是一种新型的优化计算, 它源于对鸟群觅食行为的研究, 进而利用群体智能建立的一个简化模型。在该算法中, 每个粒子就是解空间中的一个解, 粒子们根据自己的飞行经验和同伴的飞行经验来调整自己的飞行路线。群体中每个粒子在飞行过程中所经历过的最好位置, 就是粒子本身找到的最优解, 叫做个体极值 (pbest)。整个群体所经历过的最好位置, 就是整个群体目前找到的最优解, 叫做全局极值 (gbest)。群体中的每个粒子通过上述两个极值更新自己。

设粒子的群体规模为 NP , 搜索空间为 D , x_{ij}^t 表示在第 t 代第 i ($i=1, 2, \dots, NP$) 个粒子在第 j 维的位置, v_{ij}^t 表示粒子的速度, $pbest_i^t$ 表示在第 t 代时第 i 个粒子的个体极值, $gbest^t$ 表示在第 t 代时整个群体的全局极值, 每个粒子 x_{ij}^t 根据下面的公式来更新自己的速度和位置。

$$v_{ij}^{t+1} = wv_{ij}^t + c_1 r_1 (pbest_{ij}^t - x_{ij}^t) + c_2 r_2 (gbest^t - x_{ij}^t) \quad (10)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (11)$$

其中: w 为惯性权重; c_1 、 c_2 为常数, 称为学习因子; r_1 、 r_2 为 $0 \sim 1$ 之间的随机数。

3 多个邻域细胞神经网络融合 PSO 算法的边缘检测

文献[3]提出了一个有效的融合 CNN 与差分进化算法的边缘检测算子, 但是只考虑了用 1 邻域, 没有考虑多个邻域与邻域细胞之间的相互影响, 且所用种群数过多, 势必会影响程序运行速度。另外与粒子群算法相比, 遗传算法具有稳定性不高和克隆选择算法所用种群数量多等问题, 鉴于粒子群优化算法经常应用于神经网络训练, 在工程中有广泛的应用, 以及在处理高维问题其性能与差分进化算法相接近。综合以上优点, 采用粒子群优化算法代替差分进化算法, 用较少的种群规模处理高维问题, 同时考虑了多个邻域的影响, 如 2 邻域、3 邻域, 其算法流程如图 2 所示。

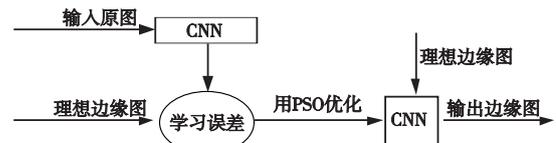


图2 算法流程

为说明该算法的有效性, 首先使用图 3(a) 所示的测验图, 其大小为 64×64 ; 图 3(b) 为其理想的边缘检测图, 可通过灰度差分获得。

迄今为止, 时间步长 (timestep) 和迭代次数 (iteration) 并没有被优化, 但是它们在 CNN 中仍然很重要。为了取得更好的结果, 如文献[3]那样优化时间步长和迭代次数, u_1 、 u_2 、 u_3 变为

$$u_1 = [a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5, z, timestep, iteration] \quad (12)$$

$$u_2 = [a_1, a_2, \dots, a_{13}, b_1, b_2, \dots, b_{13}, z, timestep, iteration] \quad (13)$$

$$u_3 = [a_1, a_2, \dots, a_{25}, b_1, b_2, \dots, b_{25}, z, timestep, iteration] \quad (14)$$

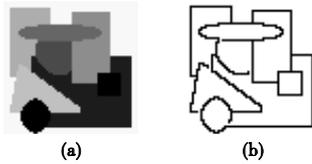


图3 测验图

算法设计的参数如下:种群大小为 20,最大迭代数为 10,惯性权重 $\omega \in [0.4, 0.9]$,线性递减,初始化搜索空间中最大速度为 10,最小速度为 -10,学习因子 $c_1 = 1.4985, c_2 = 1.27$ 。另外对于细胞神经网络中初始状态 $x_{ij}(0) = 0$,初始输入 $u_{ij}(0)$ 是输入的原图。

以 3×3 响应范围 CNN 为例(其他的可以相应给出),具体算法描述如下:

a)调用函数求解适应值。首先进行 CNN 工具箱运行环境设置,再读入图像 I ,并进行灰度差分获得理想图像 BW_1 ,分别转换成 CNN 型图像 BW_2 和 $BW_3 = [bw_{3ij}]$,则 BW_3 即为期望的输出。设置细胞神经网络运行环境中初始状态为 0,初始输入为 BW_2 ,经 CNN 运行图像 BW_2 所得到的图像 $BW_4 = [bw_{4ij}]$,即为实际输出,对 BW_4 与 BW_3 应用模板学习方法找出它们之间的对应关系,即可得到适应值,本文所采用的适应值函数为

$$\|BW_3 - BW_4\|_F = (\sum \sum (bw_{3ij} - bw_{4ij})^2)^{\frac{1}{2}}$$

b)主函数为粒子群算法,其参数设置如上所述,种群维数 13 是由式(12)决定的。式(12)中的 a_1, a_2, a_3, a_4, a_5 构成式(9)中的模板 A ,式(12)中的 b_1, b_2, b_3, b_4, b_5 构成式(9)中的模板 B ,剩余的维数分别代表阈值,时间步长 timestep 和迭代次数 iteration,将得到的模板 A 、模板 B 、阈值 z 、时间步长 timestep 和迭代次数 iteration 经 CNN 环境运行后得到 BW_4 ,计算其适应值,对每个粒子将其适应值与 pbest 作比较,如果较好,则将其作为当前值的最后位置 pbest;然后对每个粒子将其适应值与 gbest 作比较,如果较好,则将其作为当前值的好位置,注意要保证每个粒子的第 12 维为正数与第 13 维为正整数;再根据式(10)和(11)分别更新粒子的速度与位置,且惯性权重是线性递减的,循环往复进行下去,直到达到终止条件($w = 0.9 - (0.9 - 0.4) \times \text{迭代数} / \text{最大迭代数}$)。

c)用 PSO 优化算法得到的最优粒子即为所求的 CNN 克隆模板,再运行 CNN 环境显示所得到的图像。

仿真实验中,为方便与遗传算法、克隆选择算法进行比较,把遗传算法中交叉概率、变异概率、种群数、最大迭代数分别设置为 0.7、0.5、50、50,克隆选择算法的种群规模和最大迭代数与粒子群算法相同,克隆比例为 2。

4 仿真实验与结果分析

由于所涉及的 CNN 响应范围有 $3 \times 3, 5 \times 5, 7 \times 7$,不妨用范围命名该方法,即 CNN3、CNN5、CNN7 方法。为了评估该方法的有效性,用六个经典的 MATLAB 图像进行测验,并与经典的边缘检测算子(Sobel、Roberts、Prewitt、Canny 算子)和其他的优化算法(遗传算法和克隆选择算法)作比较。仿真结果如图

4~9 所示。由 CNN 理论可知邻域越多检测效果越差,因此对遗传算法和克隆选择算法只进行 1 邻域实验。

CNN3 仿真实验模板数据如下:

$$u_1 = [3.2463, 4.7036, -4.6980, 1.5527, 13.8376, 12.6404, -5.8843, 4.1217, 7.1535, 7.4331, -0.60, 152.8361, 4]$$

$$\text{即 } A = \begin{bmatrix} 3.2463 & 4.7036 & -4.6980 \\ 1.5527 & 13.8376 & 1.5527 \\ -4.6980 & 4.7036 & 3.2463 \end{bmatrix}$$

$$B = \begin{bmatrix} 12.6404 & -5.8843 & 4.1217 \\ 7.1535 & 7.4331 & 7.1535 \\ 4.1217 & -5.8843 & 12.6404 \end{bmatrix}$$

$$z = -0.6015, timestep = 2.8361, iteration = 4$$

CNN5 仿真实验模板数据用向量表示如下,即

$$u_2 = [-7.5706, 0.4510, 11.8623, 4.0636, 13.0093, -3.1146, -7.8552, -0.7618, -8.7527, -3.8907, 4.9005, 5.1761, 10.9703, 5.2886, 3.8253, 14.5763, -0.7017, 2.2664, -3.2622, 2.9064, -4.2663, 13.2661, -4.1079, 12.4116, -1.4223, 1.4287, 2.8423, 4]$$

CNN7 仿真实验模板数据用向量表示如下,即

$$u_3 = [5.5147, 4.2954, -9.2224, -9.5911, 10.5908, 12.0924, -8.5605, -0.8029, 5.7863, 7.4263, 7.3167, 14.0342, 1.3589, 1.0457, -1.1687, -5.8860, 6.8911, 8.1845, 9.3607, -4.9337, 4.2492, -5.6494, 1.6018, -7.4452, 12.5441, -3.5120, -3.5450, 10.9287, -4.3866, 11.2964, -2.1583, -3.8508, 13.8030, 13.4574, 1.6439, 6.0204, -5.2278, 11.7914, -6.2462, -5.7302, -7.4330, 0.9948, -2.0103, -1.8291, -0.8730, 3.0843, 4.7881, -7.0063, -9.0468, 0.9002, 1.9167, 2.8185, 4]$$

实验结果与分析:

a) Lena 图的边缘检测效果对比如图 4 所示。Roberts、Sobel、Prewitt 这三种算子检测效果差不多,但背景及图中人物肖像都要逊于其他算法。Canny 算子在帽子纹理、帽檐、头发还有脸部特征等方面容易发生弯曲,使得它比优化算法差很多。遗传算法和克隆选择算法在人物肖像的检测方面要明显优于经典的边缘检测算法,但是在背景以及人物的鼻子等方面的检测不清晰,比粒子群算法的检测效果要差。显然 CNN3、CNN5 和 CNN7 表现了最好的性能。

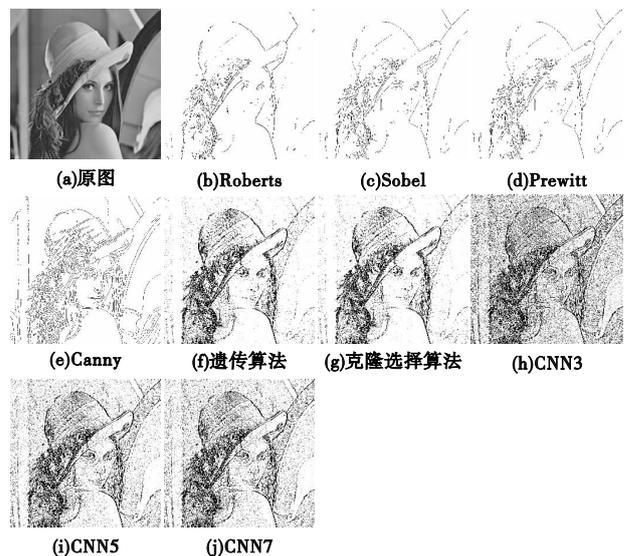


图4 Lena图像的边缘检测效果对比

b) Cameraman 图的边缘检测效果对比如图 5 所示。优化

算法与 Canny 算子对楼房、教堂、天空等的远景以及草地的检测明显优于其他算子的检测效果。但是 Canny 算子在草地的检测方面显得杂乱无章以及在楼房的检测中只能检测出其中的一栋,而优化算法却没有上述情况。此外遗传算法在人物的衣服、纽扣的检测要差于其他的优化算法。克隆选择算法与 CNN3 的检测效果相近,在人物定位以及清晰度上 CNN7 更胜一筹,这就说明研究多个邻域细胞之间的影响具有现实意义。(注意 CNN7 的背景有一些黑点,此黑点不是噪声,因为噪声不能够全是黑点,另外也注意到天空并不全是一个像素点)。

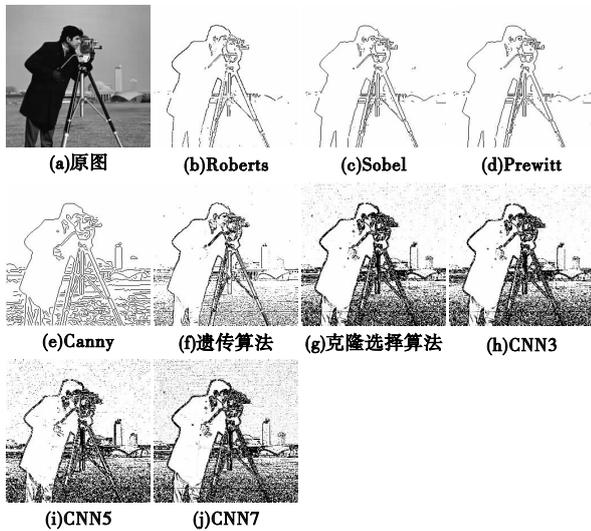


图5 Cameraman 图像的边缘检测效果对比

c) House 图的边缘检测效果对比如图 6 所示。显然优化算法及 Canny 算子的检测效果要比其他几种方法好。因为其他的几种方法在对房子的顶端检测不全,但是在房檐、窗户、砖墙及墙角等方面 Canny 算子明显不如优化算法。遗传算法与克隆选择算法相近,对于房子轮廓的检测很清晰,但是对于房子背景以及砖的排列情况检测效果差于粒子群的那三种方法,如果把图像放大可观察到呈小波浪线,而 CNN3、CNN5、CNN7 方法的效果图都能清楚地看到房子上砖的排列情况。

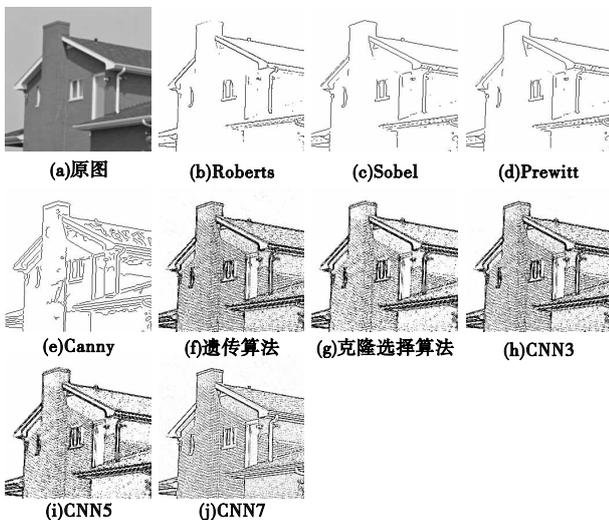


图6 House 图像的边缘检测效果对比

d) 图 7 是 texmos2.s512 的检测效果对比图。图 7 和 8 是南加州大学西皮图像数据库里很有名的纹理马赛克图,其中图 8 是 texmos3.s512。对于图 7 而言,Roberts、Sobel、Prewitt 方法检测效果相近,虽然在右下角纹理没有发生弯曲但是在上半部

分却有几条线没有检测出来。Canny 算子在左下角部分有很多的弯曲线,其余在线的连接点及其附近发生弯曲并出现不连续点。很显然优化算法就没有以上缺点。但是遗传算法在一些小的细节上检测不清,其余算法都能检测出所有纹理而且相当清晰。从效果图来看 CNN7 更胜一筹,这也说明了研究多个邻域细胞之间的影响具有重要意义。

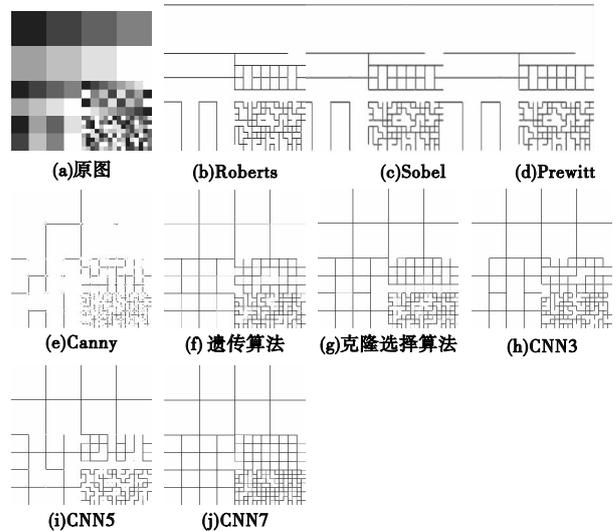


图7 texmos2.s512 图像的边缘检测效果对比

e) texmos3.s512 图的边缘检测效果对比如图 8 所示。优化算法与 Roberts、Canny 算子检测的效果要明显好于其他两个图,但是 Roberts 算子以及遗传算法检测效果差不多,都有一些断点。Canny 算子在直线上有断点和小弯曲,而其他算法却没有这些缺点,效果清晰。

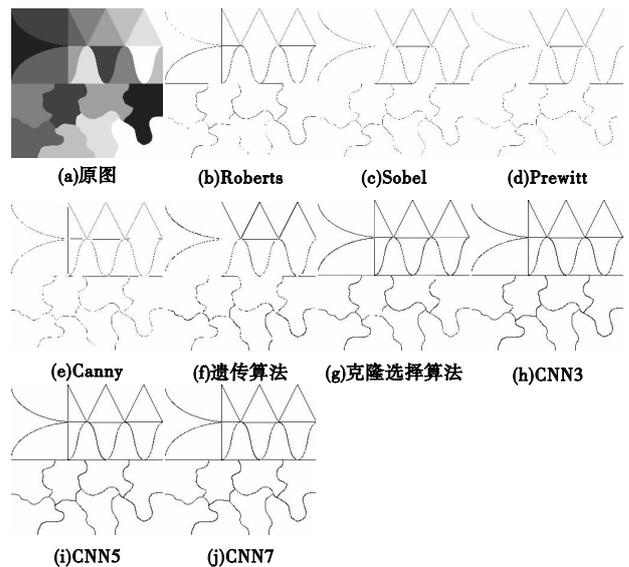


图8 texmos3.s512 图像的边缘检测效果对比

f) 图 9 是 avergra2 图,即在图 2(a) 中给出的原图经边缘检测器输出的结果。此图来源于 Aladdin v1.3,是匈牙利科学院的计算机和自动化研究所统计实验室里的图片数据库。Roberts、Sobel、Prewitt 检测效果相近,没有发生弯曲,但检测不全。Canny 算子虽然检测出一些 Roberts、Sobel、Prewitt 检测不出的线,但是有的地方却发生了弯曲,且仍有一些线检测不出。遗传算法与克隆选择算法虽明显优于以上算法,但对于图像中某些细节的检测不是很清晰,甚至出现了一些毛刺。显然不论哪一方面本文算法要明显好于其他算子,甚至在灰色三角形与黑

色之间连接的细小锯齿状及亮度不同的黑色区域都能检测出来,而且都相当清晰,这是其他边缘检测算子所办不到的。

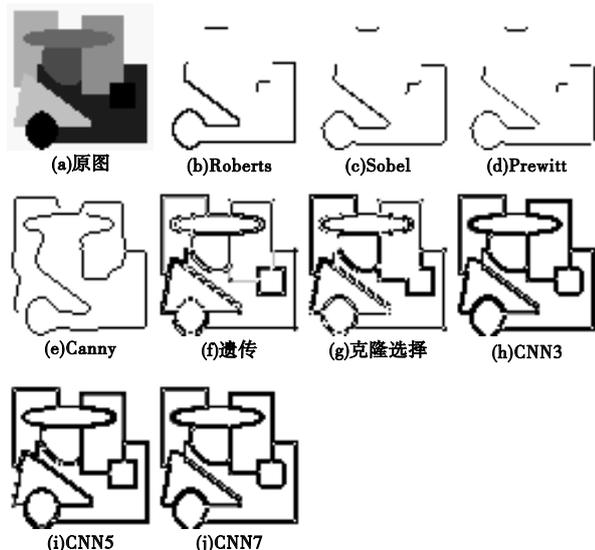


图9 avergra2图像的边缘检测效果对比

以上效果图的排列近似呈递进形式的,之所以会出现这样的排列是因为 Roberts 算子是 2×2 的模板算子,采用对角线方向相邻两像素之差的边缘检测,邻域范围小。Sobel 和 Prewitt 都是 3×3 的模板算子,所不同的是模板权重,Sobel 是先利用邻域内上下左右加权平均再进行微分运算;而 Prewitt 是 3×3 邻域内利用局部差分。因为它们都进行阈值操作,所以检测精确度不高。Canny 算子是受到广泛关注的算子,但本身也具有局限性,这是由于它对图像先进行高斯平滑处理,这就滤除了图像的细小边缘且易使边缘弯曲,又因为它采用了非极大值抑制法,这就使得它在某些方面优于 Roberts、Sobel 和 Prewitt 算子。对于遗传算法与克隆选择算法,从实验结果来看要优于经典的边缘检测算子,但是在图像边缘细节的处理上比粒子群优化算法要差一些,这主要是因为遗传算法和克隆选择算法本身的局限性,易陷入局部最优。从以上的对比分析来看本文算法具有更好的性能,这是因为本文算法不仅保留了细小边缘而且还使边缘不发生弯曲。

5 结束语

用六幅图像测试 CNN3、CNN5 和 CNN7 的边缘检测器,该检测器把 PSO 融合到了 CNN 克隆模板学习方法中。本文提出的算法比经典的边缘检测算子以及基于遗传算法和克隆选择

算法的边缘检测的基本优势是它非常高效地检测出数字图像边缘。CNN 检测器的优势在于:a)结构简单,它的 transfer 函数有分段线性表现;b)边缘检测非常高效,而且检测性能的提高取决于优化算法的有效性;c)CNN 算法易于实现。

细胞神经网络可被作为简单但强大的工具用于数字图像的高效边缘检测。由于本文提出的算法没有加入噪声的影响,如果加入噪声(如文献[11])会使结论更加圆满,这也是下一步要开展的研究工作。

参考文献:

- [1] MALLAT S, ZHONG S. Characterization of signals from multiscale edges[J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 1992, 14(6): 710-732.
- [2] ZHAO Jian-ye, WANG Hai-ming, YU Dao-heng. A new approach for edges detection of noisy image based on CNN[J]. *International Journal of Circuit Theory Applications*, 2003, 31(2): 119-131.
- [3] BASTÜRK A, ENIS G. Efficient edge detection in digital images using a cellular neural optimized by differential evolution algorithm[J]. *Expert Systems with Application*, 2009, 36(2): 2645-2650.
- [4] GOLDBERG D E. Genetic algorithms in search, optimization, and machine learning[M]. Boston: Addison-Wesley, 1989.
- [5] KIM J, BENTLEY F J. Immune memory in the dynamic clonal selection algorithm[C]//Proc of the 1st International Conference of Artificial Immune System. 2002.
- [6] CHUA L O, YANG L. Cellular neural networks: applications[J]. *IEEE Trans on Circuits and Systems-I: Fundamental Theory and Applications*, 1988, 35(10): 1273-1290.
- [7] CHUA L O, ROSAK T. The CNN paradigm[J]. *IEEE Trans on Circuits and System-I: Fundamental Theory and Applications*, 1993, 40(3): 147-156.
- [8] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proc of IEEE International Conference on Neural Networks. 1995.
- [9] HU X, EBERHART R, SHI Y. Engineering optimization with particle swarm[C]//Proc of IEEE Swarm Intelligence Symposium. 2003.
- [10] ZARANDY A. The art of CNN template design[J]. *International Journal of Circuit Theory and Applications*, 1999, 27(1): 5-23.
- [11] LI Hua-qing, LIAO Xiao-feng, LI Chuan-dong, et al. Edge detection of noisy images based on cellular neural network[J]. *Commun Nonlinear Sci Numer Simulat*, 2011, 16(9): 3746-3759.
- [12] CHUA L O, YANG L. Cellular neural networks: theory[J]. *IEEE Trans on Circuits and Systems-I: Fundamental Theory and Applications*, 1988, 35(10): 1257-1272.