

第七届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: 2014@tzmcm.cn

七届“认证杯”数学中国

数学建模网络挑战赛

论文格式规范

- 论文（答卷）用白色 A4 纸，上下左右各留出 2.5 厘米的页边距。
- 论文第一页为承诺书，具体内容和格式见本规范第二页，该页由队员及教练员（如有）手写签字后生效（**一定要注明参赛组别，挑战赛组委会将各组别分开评阅**）。
- 论文第二页为编号专用页，用于评委团评阅前后对论文进行编号，具体内容和格式见本规范第三页。
- 论文题目和摘要写在论文第三页上，从第四页（若无英文摘要）开始是论文正文。
- 论文第一页为承诺书，论文第二页为编号专用页，用于评委团评阅前后对论文进行编号。论文题目、关键词（模型、算法名称）和摘要写在论文第三页上，第四页为英文摘要（选填），论文 1—4 页按组委会统一要求编排，具体内容见下文，从第五页开始是论文正文。论文从正文开始编写页码，页码必须位于每页页脚中部，用阿拉伯数字从“1”开始连续编号，**注意，论文一律要求从左侧面装订**。
- 论文必须有页眉，页眉标识参赛队号，例如队号为 1101 的队需标识：**参赛队号 # 1101**
- 论文中不能有任何可能显示答题人身份的标志。
- 论文题目用三号黑体字、一级标题用四号黑体字，并居中。论文中其它汉字一律采用小四号黑色宋体字，行距用单倍行距。图，表需有标题，并且编号。核心公式要编号。
- 提请大家注意：摘要在整篇论文评阅中占有重要权重，请认真书写摘要（注意篇幅不能超过一页）。评委团评阅时将首先根据摘要和论文整体结构及概貌对论文优劣进行初步筛选。
- 引用别人的成果或其他公开的资料（包括网上查到的资料）必须按照规定的参考文献的表述方式在正文引用处和参考文献中均明确列出。正文引用处用方括号标示参考文献的编号，如 [1][3] 等；引用书籍还必须指出页码。参考文献按正文中的引用次序列出，其中书籍的表述方式为：
[编号] 作者，书名，出版地：出版社，出版年。
参考文献中期刊杂志论文的表述方式为：
[编号] 作者，论文名，杂志名，卷期号：起止页码，出版年。
参考文献中网上资源的表述方式为：
[编号] 作者，资源标题，网址，访问时间（年月日）。
- 论文中使用到的程序源代码放在附录中给出。
- 本规范的最终解释权属于数学建模网络挑战赛组委会所有。

数学建模网络挑战赛组委会
2014 年 5 月

第七届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: 2014@tzmcm.cn

第七届“认证杯”数学中国

数学建模网络挑战赛 承 诺 书

我们仔细阅读了第七届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为：#2785

参赛队员（签名）：

队员 1：陈敏

队员 2：侯雯琰

队员 3：彭芳

参赛队教练员（签名）： 李春萍

参赛队伍组别(中学|专科|本科|研究生)： 本科

第七届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: 2014@tzmcm.cn

第七届“认证杯”数学中国

数学建模网络挑战赛 编号专用页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：#2785

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

第七届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: 2014@tzmcm.cn

2014 年第七届“认证杯”数学中国 数学建模网络挑战赛第二阶段论文

题 目 基于双线性插值与边缘检测的图像

放大算法的研究与实现

关 键 词 双线性插值 边缘检测 BP 神经网络 质量评价体系

摘 要：

科技发展要求图像放大技术不能仅仅局限于放大倍数的多少，更多关心的是如何在提高放大倍数的同时保证图像质量，即要求图像精度满足一定的使用要求。本文将图像放大过程分为内部图像插值放大与边缘图像插值放大两个部分，重点研究基于双线性图像内部插值与 BP 神经网络图像边缘插值放大问题。

首先，综合线性插值与非线性插值方法，得到最优双线性插值法。借助 Matlab 对原始图像进行加载、二值化处理、整数像素边缘检测、亚像素边缘检测、图像插值放大和图像保存等

其次，由于边缘曲线灰度梯度变化较大，需要精细的插值算法，本文选取了基于亚像素边缘检测的 BP 神经网络模型进行多次节点选值、样本训练得到精细化边缘。

然后，建立图像放大质量评价体系，利用 Matlab、Excel 等软件，从主观视觉感受和客观数据分析两方面评价图像放大质量，计算放大后图像的峰值信噪比、均方误差和平均梯度值进行分析，发现本文提出的综合算法对马赛克现象有明显减弱，边缘更加清晰，放大效果较好。

最后，对模型的优缺点进行总结，另外对二维彩色低分辨率图像算法进行三维图像的拓展，深入思考。本文分析思路清晰，切入点独到，运用多种插值方法分析全面，特点鲜明。

参赛队号： #2785

所选题目： B 题

参赛密码 _____
(由组委会填写)

第七届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会
电话：0471-4969085

邮编：010021

网址：www.tzmcm.cn
Email: 2014@tzmcm.cn

英文摘要（选填）

（此摘要非论文必须部分，选填可加分，加分不超过论文总分的 5%）

Many technology development requirements of image magnification technique is not limited to the magnification, more concerned about is how to ensure the quality of the image at the same time improve magnification, which requires the image accuracy satisfy certain requirements. This paper will image amplification process is divided into internal image magnification of two parts and the edge of image interpolation, bilinear interpolation image focuses on the interior and edge BP neural network interpolation problem.

First of all, integrated linear interpolation and nonlinear interpolation method, the optimal bilinear interpolation. With the help of Matlab loading, the original two value of the image processing, the integer pixel edge detection, subpixel edge detection, image interpolation and image storage

Secondly, because of the edge curve of gray gradient change is bigger, need fine interpolation algorithm, this paper chooses the BP neural network model based on sub pixel edge detection for multiple nodes selected value, samples are trained to obtain fine edge. Then, a magnified image quality evaluation system, using Matlab, Excle and other software, from subjective visual perception and objective data analysis of two aspects of quality evaluation of image magnification, calculation of magnified image peak signal to noise ratio, mean square error and the gradient analysis, found that the comprehensive algorithm is proposed in this paper has significantly weakened the mosaic phenomenon the edge, more clear, amplification effect.

Finally, the advantages and disadvantages of the model are summarized, in addition to the 2D color images of low resolution algorithm to expand the three-dimensional image, deep thinking. In this paper, the analysis of clear thinking, the starting point, using a variety of interpolation method of comprehensive analysis, characteristics.

Key Words: Bilinear interpolation Edge detection BP neural network Quality evaluation system

目 录

§ 1 问题的重述.....	1
1.1 背景.....	1
1.2 问题分析以及整体思路算法流程.....	1
§ 2 模型假设.....	2
§ 3 名词解释.....	2
3.1 名词解释.....	2
§ 4 符号说明.....	3
§ 5 模型的建立与求解.....	3
5.1 内部插值放大图像算法模型.....	3
5.2 神经网络细化图像边缘算法.....	7
5.3 图像质量评价体系.....	12
5.4 实验结果分析.....	13
§ 6 模型的误差分析.....	17
§ 7 模型的评价.....	18
7.1 模型的优点:	18
7.2 模型的缺点.....	18
§ 8 模型的改进与推广.....	18
8.1 模型改进.....	18
8.2 模型的推广.....	18
§ 9 参考文献.....	20

§ 1 问题的重述

1.1 背景

目前图像放大处理已经被广泛用于航空航天、军事保障、医学成像和虚拟开发等多个领域中。图像放大的本质是图像像素点和像素灰度信息的增加，实际上就是图像数据量的扩充，因此图像放大算法主要解决的问题就是怎样扩展图像像素点，包括增加信息的位置和扩充信息的数值。

科技发展要求图像放大技术不能仅仅局限于放大倍数的多少，更多使用者关心的是如何在提高放大倍数的同时保证图像质量，即要求图像精度满足一定的使用要求。

图像边缘和图像内部信息无论在表现形式或是信号特性上均有较大差异，而这种细节差异恰恰是精确图像处理所需要关注的问题，由于内部梯度变化不明显，对于大多数图像插值算法来说，都是适用的。常用的图像放大插值算法依据其插值过程可以分为线性插值和非线性插值。线性方法对不同位置概念的图像信息进行相同处理，处理速度快，但易产生边缘模糊性或锯齿性，对细节描述能力较差。非线性插值方法针对图像不同位置处理方法相异，但复杂度的增加往往带来较大的计算量。边缘部分灰度梯度变化较大，所以需要特定的插值算法实现放大。因此可以说图像边缘处理后的清晰程度是决定图像精度的根本要素。在确定图像内部信息的基础上如果可以精确确定图像边缘，针对边缘采取相应手段进行特殊插值，就可以从根本上实现图像的精细放大，改善图像的最终显示效果，这种图像放大处理机制对于图像处理问题的研究是很有益处的。

1.2 问题分析以及整体思路算法流程

本文正是从图像边缘和图像内部信息考虑，使用组合的方法精细放大图像，得到较好的图像效果。分析传统图像插值算法的特点，图像内部选择线性插值、非线性插值，图像边缘运用亚像素曲线拟合定位提取边界坐标，再利用基于神经网络的插值算法进行放大。

具体流程如下：

1. 基于线性插值算法和非线性插值算法对图像内部进行差值放大处理，依据图像放大原理，针对运算模型通过加权平均算法进行相应信息补充，使用最近邻插值法、双线性插值法和双立方卷积插值以及小波插值补充图片内部邻接点像素值。

2. 接着对于图像边缘锯齿现象进行细化，使用亚像素边缘检测提取坐标，BP 神经网络放大算法细化边缘得到更准确的定位。BP 神经网络以多层感知基础上实现网络泛化训练，即确定各子细胞单元的相对阈值和权值 ω_n ，设置节点确定像素点灰度值。

3. 建立图像质量评价体系，通过直观评价（直接观察），客观评价（峰值信噪比(PSNR)，均方误差(MSE)，平均梯度值)进行效果评价，得到最佳图片放大方法。

4. 对整个模型进行结果分析。

具体方案流程图如图 1 所示。

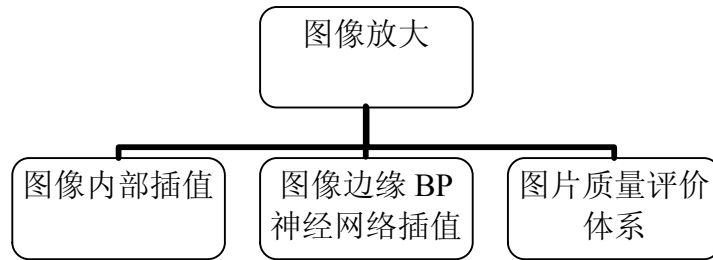


图 1 方案流程图

§ 2 模型假设

- 1、本文建立的模型针对分辨率较低的图形（分辨率高的话，进行放大也不会出现如题中所说的可能产生的马赛克效果）。
- 2、假设对图像进行边缘提取的时候没有出现大量的数据丢失。
- 3、假设图片内部颜色变化不大。

§ 3 名词解释

3.1 名词解释

1. 低分辨率单元：由四个原始图像中的低分辨率像素点 $((i, j), (i, j+1), (i+1, j), (i+1, j+1))$ 组成的一个小正方形为一个低分辨率 (LowResolution, LR) 单元元。

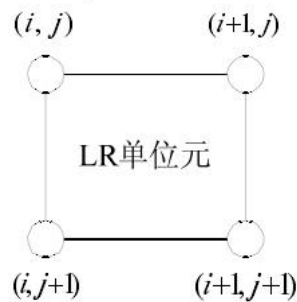


图 2 低分辨率单元

2. 双线性插值：又称为双线性内插。在数学上，双线性插值是有两个变量的插值函数的线性插值扩展，其核心思想是在两个方向分别进行一次线性插值。

3. 边缘检测：边缘检测是图像处理和计算机视觉中的基本问题，边缘检测的目的是标识数字图像中亮度变化明显的点。图像边缘检测大幅度地减少了数据量，并且剔除了可以认为不相关的信息，保留了图像重要的结构属性。

4. BP 神经网络：BP (Back Propagation) 神经网络是一种按误差逆传播算法训练的多层前馈网络，是目前应用最广泛的神经网络模型之一。BP 网络能学习和存贮大量的输入-输出模式映射关系，而无需事前揭示描述这种映射关系的数学方程。它的学习规则是使用最速下降法，通过反向传播来不断调整网络的权值和阈值，使网络的误差平方和最小。

5. 均方误差 (MSE)：MSE 可以评价数据的变化程度，MSE 的值越小，说明预测模型描述实验数据具有更好的精确度，用来反映放大图像与原图的逼近程度。

6. 峰值信噪比 (PSNR)：是一种评价图像的客观标准 and 一种评鉴画质的客观量测方法。它是原图像与被处理图像之间的均方误差相对于 $(2^n - 1)^2$ 的对数值 (信号最大值的

平方， n 是每个采样值的比特数)，

§ 4 符号说明

序号	符号	符号说明
1	r_x	放大系数
2	$g(i, j)$	图像像素的值
3	$\psi(t)$	基本小波函数
4	E_i	梯度算子
5	ω_{rt}	BP神经网络层之间的连接权值
6	(x^{sp}, y^{sp})	亚像素边缘点二维位置坐标
7	$\alpha(t)$	学习率
8	MSE	均方误差
9	$PSNR$	峰值信噪比

§ 5 模型的建立与求解

5.1 内部插值放大图像算法模型

要得到放大后清晰的图像效果就首先要了解图像放大原理。

5.1.1 图像放大原理

通过将原图像的坐标 (x, y) 乘以放大系数 r_x 和 r_y 。产生新的坐标 (x', y') ，变换表达式为 $x' = x * r_x$ ， $y' = y * r_y$ 。以矩阵的形式表示为

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} r_x & 0 & 0 \\ 0 & r_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

其中， $r_x > 1$ ， $r_y > 1$ 分别为 x 、 y 方向的放大比例。

当对一个图像进行放大时（如放大 4 倍），首先通过放大的倍数得到新图像的高度和宽度，此时新图像是一个待填充的空白图像网格。然后依据原图像中已知像素点的值，按照放大倍数找到它们在新图像中的位置并直接进行填充，即像素 $(4i, 4j)$ 的值直接取自原图像。但是所产生的空白像素点，其值是未知的（如点 p ），这时可根据已知像素点的值采用插值算法进行估算，逐步填充空白网格直至完整，从而完成图像放大的处理。如图 3 所示。

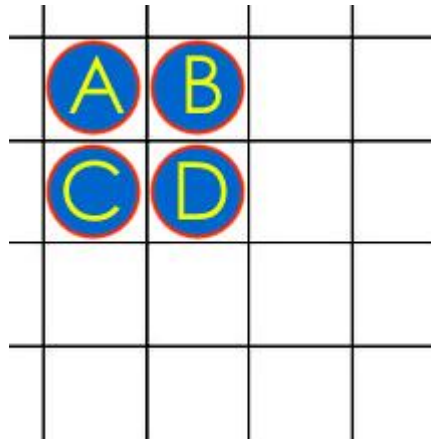


图 3 原始图像相邻的 4 个像素点分布图

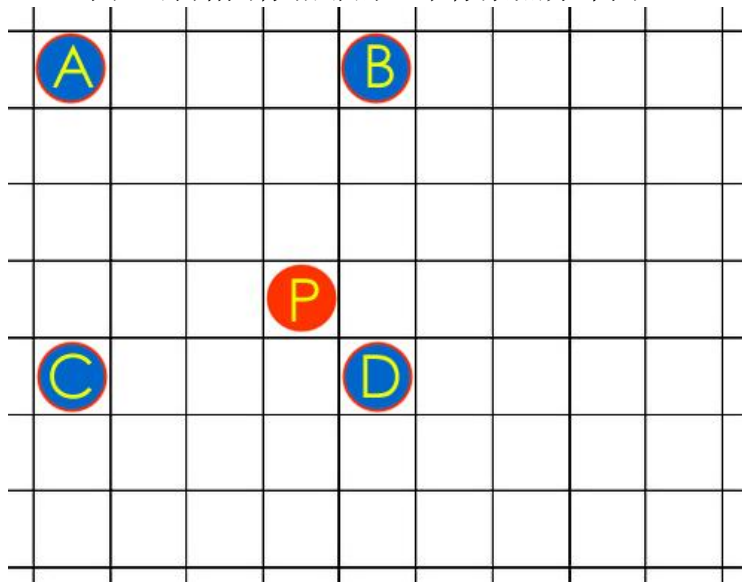


图 4 图像放大 4 倍后已知像素分布图

5.1.2 线性插值

根据图像放大原理，以下通过传统的线性插值法进行图像内部处理，经典线性插值方法有：a. 最近邻插值法 b. 双线性插值法 c. 双立方卷积插值。

a. 最近邻插值法

放大后未知的像素点 P ，将其位置换算到原始影像上，与原始的邻近的 4 周像素点 A, B, C, D 做比较，令 P 点的像素值等于最靠近的邻近点像素值。如上图中的 P 点，由于最接近 D 点，所以就直接取 $P=D$ 。

以上算法可以保持像素灰度的有效复制，在灰度变化不大的情况下有效地完成像素扩充；但存在当图像灰度变化程度较高时，在像素点中点处的像素值会突然出现一个跳跃，出现马赛克和锯齿等明显走样的现象，造成插值失真较为严重，插值结果粗糙结果。

b. 双线性插值算法

将放大后未知的像素点 P ，其位置换算到原始图像上，计算一个低分辨率单元的四个像素点 A, B, C, D 对 P 点的影响(越靠近 P 点取值越大，表明影响越大)，利用需要处理的原图像素点周围的 4 个邻点的像素值在 x, y 两个方向上作线性内插，即待插点 (u_0, v_0) 的最终灰度值并非由其最近距离的单一像素点唯一确定，而是取决于 (u_0, v_0) 附近多个整数点，用离待插点最近的四个点的值加权求得。设置坐标通过向后映射法得到浮点坐标 $(i+u, j+v)$ 。其中 i, j 是非负整数， u, v 是取值 $[0,1)$ 区间的浮点数，随放大后像素行

列坐标的不同而改变。则这个像素的值 $g(i+u, j+v)$ 可由一个低分辨率单元周围四个像素的值决定，公式为

$$g(i+u, j+v) = (1-u)(1-v)g(i, j) + (1-u)vg(i, j+1) + u(1-v)g(i+1, j) + uv g(i+1, j+1) \quad (2)$$

其中， $g(i, j)$ 表示源图像 (i, j) 处的像素值，以此类推。原理如图 6 所示。

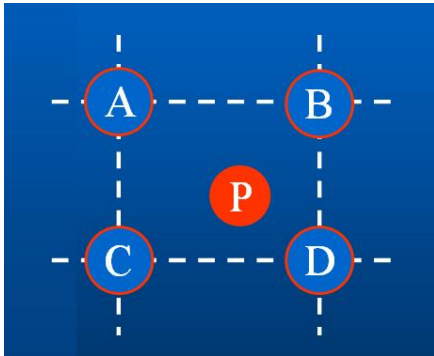


图 5 4 个邻近点影响未知点信息

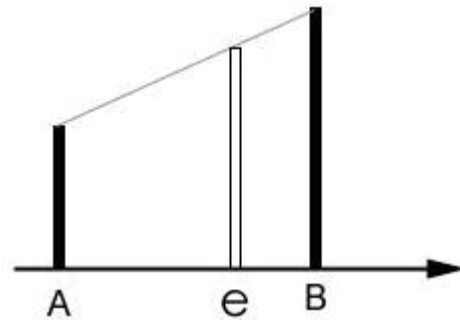


图 6 线性插值算法求值示意图

在水平和垂直两个方向上均进行插值，分别沿着 X 轴和 Y 轴完成两次计算，计算过程中循环对图像每行和每列的数据进行计算，不同行列间的计算过程彼此独立不依赖于执行顺序，且对于 AB 两像素点之间的其它像素点的色彩值，认定为直线变化的，要求 e 点处的值，找到对应位置直线上的点即可。

双线性插值图像放大算法的实现步骤如下：

- (1) 读入原图像，得到图像的高度和宽度；
- (2) 根据放大倍数得到新图像的高度和宽度，创建空白新图像；
- (3) 利用双线性插值得到像素点 (x', y') 即 e, f 点再计算出对 p 点的影响值写回新图像；
- (4) 得到放大图像。其中第 (3) 步，即双线性插值部分，输出像素的值只由输入像素的邻域像素值确定，后一像素输出结果不依赖于前面像素处理的结果。且输入和输出图像分别利用两个数组来进行存储。

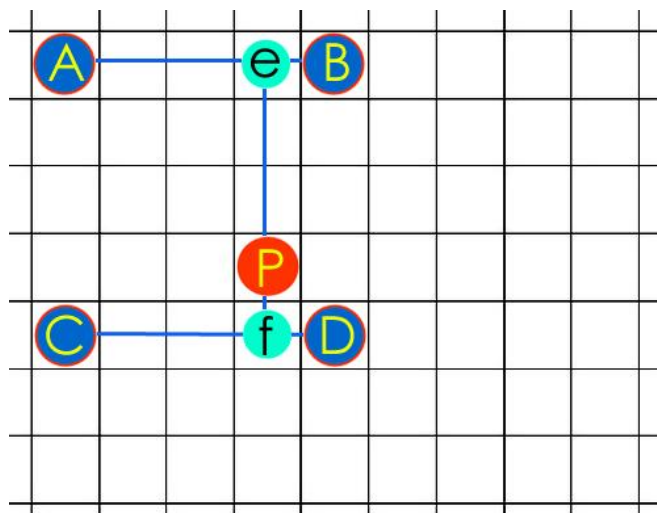


图 7 双线性插值法示意图

c. 双立方插值算法

对于放大后未知的像素点P，将对其影响的范围扩大到邻近的16个像素点，依据对P点的远近影响进行插值计算，因P点的像素值信息来自16个邻近点，所以可得到较细致的影像。

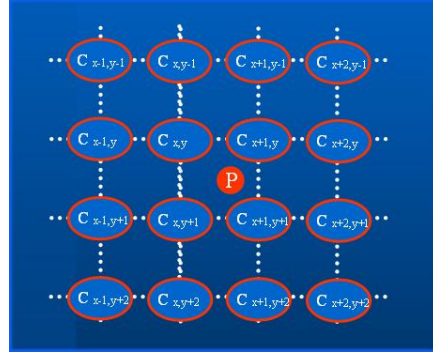


图8 16个邻近点影响未知点信息

5.1.3 非线性插值法

由于上述线性插值的本质是通过滤波器的低通技术实现插值运算，但低通滤波显然会丢失大量重要的高频信号，产生很多不必要的失真；为了克服线性插值带来的这些问题，利用非线性技术实现图像的精确放大处理。典型的非线性插值方法有(a)分形插值；(b)小波插值；(c)偏微分插值算法。这里以小波插值算法为例进行简单介绍。

小波插值算法具体步骤为：

(1)小波变换原图像

如果函数 $\psi(t)$ 满足

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \quad (3)$$

$$\int_{-\infty}^{\infty} |\hat{\psi}(\omega)|^2 \cdot |\omega|^{-1} d\omega = C_{\psi} < \infty \quad (4)$$

其中， $\hat{\psi}(\omega)$ 为 $\psi(t) \parallel \Delta X \parallel$ 的 Fourier 变换。则 $\psi(t)$ 为基本小波函数。

通过相应计算得到一组小波函数簇

$$\psi_{ab}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right) \quad (5)$$

定义小波变换

$$(W_{\psi}f)(a,b) = \int_{-\infty}^{\infty} f(t) \cdot \overline{\psi_{ab}(t)} dt = |d|^{1/2} \int_{-\infty}^{\infty} f(t) \cdot \overline{\psi\left(\frac{t-b}{a}\right)} dt \quad (6)$$

(2)通过计算变换系数等级可以作为插值处理图像细节系数和近似系数的估算标准，最终完成图像插值。

5.2 神经网络细化图像边缘算法

以上算法实现对图像内部放大，但在图像边缘处理缺乏更精确的定位，所以本文提出通过 BP 神经网络边缘放大算法细化边缘得到更准确的定位。

5.2.1 图像边界亚像素提取和

亚像素曲线拟合定位方法如下

(1) 曲线拟合亚像素定位法是在完成整数像素级边缘检测的基础上进行处理的，取任一整数级边缘点为模板中心，选择 5×5 邻域模板，计算窗口内各像素点的灰度差分，通过比较计算差分值，确定图像灰度梯度方向和边缘方向；

(2) 对梯度差之和最小方向上的灰度差进行曲线拟合，得到其函数方程，并对该

函数求导，导数极值位置就类等于整数像素点对应的亚像素边缘位置；

(3) 重复步骤 (1) 和步骤 (2)，对原始整数像素边缘点逐一进行处理，直至边缘集合为全集。

综上，曲线拟合定位方法可以分为：边缘方向确定，二次曲线拟合和边缘点坐标转换三个步骤。

5.2.2 边缘方向的确定

边缘方向的确定是在完成整数像素级边缘检测的基础上实现的，其中邻域阶数的选择尤为重要，阶数过高会增加运算的复杂程度，影响算法的实时性；阶数过低，会使灰度梯度方向难以确定，影响边缘检测的效果，通过多区域试验研究表明， 5×5 的区域既可以保证算法运算效率又可以在一定程度上降低区域灰度梯度确定的难度，因此 5×5 的计算窗口是本文边缘方向确定的区域。

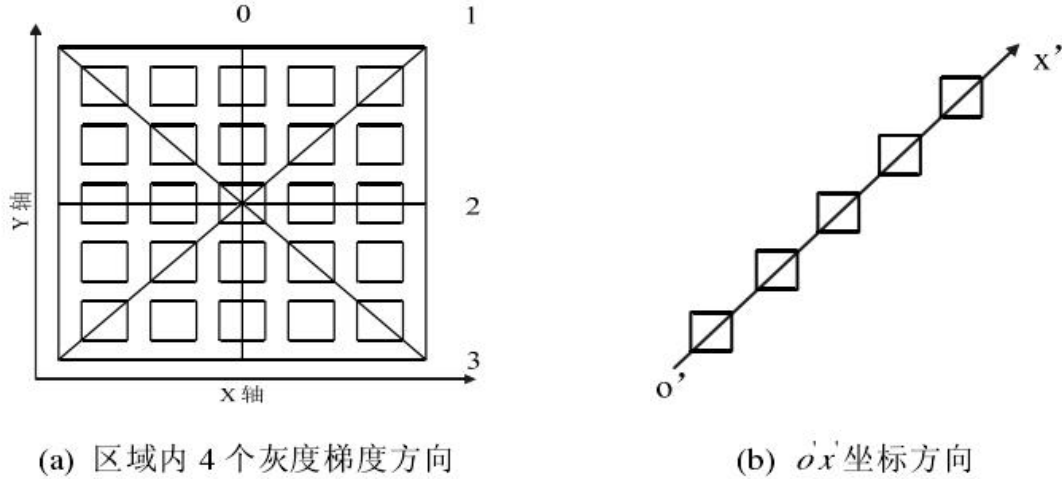


图 9 区域灰度梯度方向的不同坐标系表示

在上述 5×5 区域内建立坐标系 xoy ，并在此图像坐标系内定义如图 9(a) 所示的 4 个方向，分别用方向 0、方向 1、方向 2 和方向 3 表示，各方向依次相差 45° ，任一方向上相邻两像素点之间的灰度差值 $p_{n+1} - p_n$ ，定义 E_i 为梯度算子，计算表达式如下：

$$E_i = \sum_{n=0}^4 |p_{n+1} - p_n| \quad (7)$$

其中 $i = 0, 1, 2, 3$ 。由梯度算子原理可知， E_i 保持最小的变化方向为边缘方向，而灰度梯度方向与边缘方向相垂直，如图 9(b) 建立灰度梯度方向坐标系 $o'x'$ 。

5.2.3、二次曲线拟合

5.2.2 节中已经完成了从区域内 4 方向中确定图像边缘灰度梯度方向 $o'x'$ ，在此基础上，试完成二次曲线拟合，拟合依据是 $o'x'$ 坐标方向上的像素坐标大小和相应的顺序灰度差值。设二次曲线表达式为 $f(x') = Ax'^2 + Bx' + C$ 完成二次曲线的设计就是确定 A、B 和 C 这 3 个未知系数的大小。

在 $o'x'$ 坐标方向上任取 3 个相邻像素点及其相应灰度插值 $x'_j (j=1,2,3)$ 通过此 6 个已知数据，联立求解二次曲线 $f(x') = Ax'^2 + Bx' + C$ 的未知相关系数：

$$\begin{cases} A = \frac{[f(x'_3) - f(x'_1)](x'_2 - x'_1) - [f(x'_3) - f(x'_1)](x'_3 - x'_1)}{(x'_2 - x'_1)(x'^2_3 - x'^2_1) - (x'^2_2 - x'^2_1)(x'_3 - x'_1)} \\ B = \frac{f(x'_3) - f(x'_1) - (x'_3 - x'_1)A}{(x'_3 - x'_1)} \\ C = f(x'_1) - x'^2_1 A - x'_1 B \end{cases} \quad (8)$$

曲线的导数 $f'(x'_j)$ 为:

$$f'(x'_j) = 2Ax'_j + B \quad (9)$$

因此最终求取的亚像素边缘点坐标在 $o'x'$ 坐标系中的位置就是 $f'(x'_j) = 0$ 处位置, 即满足:

$$f'(x'_j) = 2Ax'_j + B = 0 \Rightarrow x'_{j\max} = -\frac{B}{2A} \quad (10)$$

5.2.4、边缘点坐标转换

5.2.3 节求取的 $x'_{j\max}$ 可以表示为式 (11) 的形式:

$$x'_{j\max} = x'^p_{j\max} + x'^{sp}_{j\max} \quad (11)$$

其中 $x'^p_{j\max}$ 表示 $x'_{j\max}$ 的整数部分, $x'^{sp}_{j\max}$ 表示 $x'_{j\max}$ 的小数部分。设 (x, y) 为 $x'^p_{j\max}$ 对应的图像坐标系中整数像素级坐标位置; (x^{sp}, y^{sp}) 为 $x'^{sp}_{j\max}$ 对应的图像坐标系中亚像素级坐标位置, 下面分 3 种情况讨论不同的坐标方向上边缘点坐标转换为图像坐标系的结果。

1) 灰度梯度 0 方向:

$$\begin{cases} x^{sp} = x \\ y^{sp} = y + x'^{sp}_{j\max} \end{cases} \quad (12)$$

2) 灰度梯度 2 方向:

$$\begin{cases} y^{sp} = y \\ x^{sp} = x + x'^{sp}_{j\max} \end{cases} \quad (13)$$

3) 灰度梯度 1 或 3 方向:

$$\begin{cases} x^{sp} = x + x'^{sp}_{j\max} \times \cos 45^\circ \\ y^{sp} = y + x'^{sp}_{j\max} \times \sin 45^\circ \end{cases} \quad (14)$$

通过分析上述 3 种情况分析, 我们最终得到图像边缘检测中亚像素级坐标精确定结果, 通过 **Matlab** 仿真, 可以得到如图 10 所示的亚像素边缘定位效果图。

部分代码如下

```
for y=2:1:n-1
    if f(x,y)==0
        W(x,y)=0;
    else
        i=[R(x-1,y),R(x,y),R(x+1,y)];
        j=[R(x,y-1),R(x,y),R(x,y+1)];
        if (i(2)>i(1))&(i(2)>i(3))&(j(2)>j(1))&(j(2)>j(3))
            Xe=x+(i(1)-i(3))/((i(1)-2*i(2)+i(3))+eps); %求亚像素边缘坐标[Xe,Ye]
            Ye=y+(j(1)-j(3))/((j(1)-2*j(2)+j(3))+eps);
            f(Xe<m)&(Xe>0)&(Ye<n)&(Ye>0)
```

参赛队号#2785

```

Xe1(nn)=Xe; %得到的亚像素边缘点
Ye1(nn)=Ye;
Xe2(nn)=round(Xe); %得到的逼近的边缘点
Ye2(nn)=round(Ye);
x1=Xe2(nn);
y1=Ye2(nn);
A(nn)=x1;
B(nn)=y1;
W(Xe2(nn),Ye2(nn))=1; %把检测到的边缘点直观的表达出来
nn=nn+1; %设置循环，用数组的形势表示 Xe2, Ye2, 方便后期查看处
理亚像素边缘点坐标
end

```

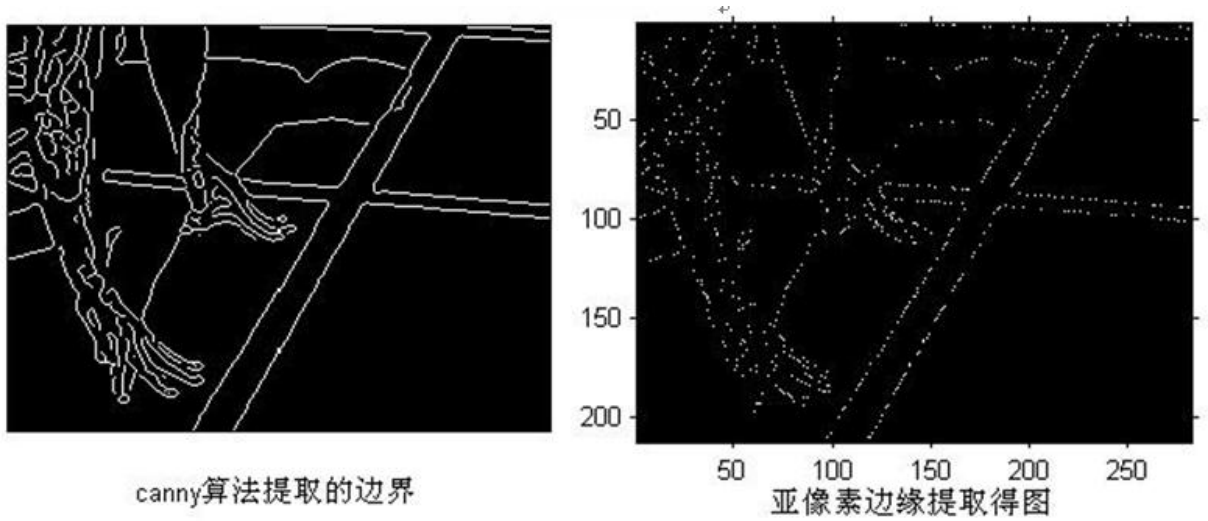


图 10 亚像素边缘定位图

5.2.5 BP 神经网络的核心思想

BP 神经网络以多层感知机描述作为核心思想，整个网络架构由输入层 (Input Level)、隐层 (Hidden Level) 和输出层 (Output Level) 组成，输入层、隐藏层和输入层之间存在连接权 ω_{ri} 、 ω_{ri} 反映不同层级神经细胞之间的连接强度。神经网络算法基础就是网络泛化训练，即确定各子细胞单元的相对权值 ω_{ri} ，具体步骤如下。

(1) Input Level 各神经单元与 Hidden Level 各神经单元之间的关联权重 $V_{hi} (h=1,2,...,n; i=1,2,...,p)$ ，Hidden Level 各神经单元与 Output Level 各神经单元之间关联权重为 $W_{ij} (i=1,2,...,p; j=1,2,...,q)$ ；令 n 为 Input Level 节点数， p 为 Hidden Level 节点数， q 为 Output Level 节点数。

(2) 对样本 $(X_k, Y_k) (k=1,2,...,m)$ 进行操作如下。

1) 样本数 X_k 进入 Input Level 单元，与连接权 V_{hi} 计算作用，得到 Hidden Level 单元学习激活的新结果

$$b_i = f\left(\sum_{h=1}^n V_{hi} x_h + \theta_i\right) \quad i=1,2,...,p \quad (15)$$

其中 f 为 S 型函数 $f(x) = (1 + e^{-x})^{-1}$ ， θ_i 为偏移。

2) 计算 Output Level 单元激活值。

$$y_j = g\left(\sum_{i=1}^p W_{ij} b_i + \theta_j\right) \quad i=1,2\dots q \quad (16)$$

其中 g 也选用 S 型函数 $f(x) = (1 + e^{-x})^{-1}$, θ_j 为偏移。

3) 依据原始样本 Y_k , 计算 Output Level 单元误差。

$$E(\omega) = \sum_{i=1}^q \|y_i - y_i^k\|^2 \quad (17)$$

y_i^k 为单元 i 的期望输出。

4) 判断 $E(\omega)$ 与给定误差 ε 关系。

如果 $E(\omega) < \varepsilon$, 结束训练;

如果 $E(\omega) \geq \varepsilon$, 反向传播;

权值调整方法如下

$$\Delta W_{ij}^{(k-1)}(t+1) = \eta \bullet \Delta W_{ij}^{(k-1)}(t) + (1-\eta) \bullet G_{ij}^{(k-1)}(t) \quad (18)$$

$$G_{ij}^{(k-1)}(t) = \frac{\partial E}{\partial \Delta W_{ij}^{(k-1)}(t)} \quad (19)$$

其中 $0 < \eta < 1$ 为网络调整的能动因子; $\alpha(t)$ 为学习率。

(3) 重复步骤(2), 直到样本集合 (X_k, Y_k) 误差满足需求, 此时的神经网络被称为自适应学习训练网络, 如果此时将新样本的输入值 X 引入已经完成训练的 BP 网络, 就可以得到具有训练相似效度的有效输出值 Y 。

5.2.6 BP 神经网络结构

利用原始亚像素边缘检测点量化像素灰度的首要问题是 BP 网络模型的整体设计, 从四个方面着手, 分别是输入输出层节点选取, 隐层节点设计, 实验样本数据的预处理和学习率的选取。

(1) 输入输出层的节点设计

采用 BP 神经网络实现图像插值放大, 根本任务是根据任意设定的像素点位置, 计算出该点的灰度大小。

为了易于神经网络收敛, 直接将像素点灰度大小 h^{sp} 作为 BP 模型的目标参数, 因此 Output Level 节点数为 1, 而亚像素边缘点二维位置坐标 (x^{sp}, y^{sp}) 作为输入, 由此可以确定网络对应的 Input Level 节点数为 2。

$$N = \sqrt{n + p + a} \quad (20)$$

其中 α 为 $0 \sim 10$ 之间常数, 从计算精确和效率等方面着手考虑后, 本文确定隐层节点数 $N=4$ 。

(2) 样本数据的预处理。

依照式(21)对输入、输出样本参数进行 $0 \sim 1$ 范围内的归一化处理, 从而避免网络学习过程中 Hidden Level 到 Input Level 之间, Output Level 到 Hidden Level 之间权值调节的困难, 提高网络的收敛速度和精度。

$$x = \frac{0.8}{x_{\max} - x_{\min}} (x - x_{\min}) + 0.1 \quad (21)$$

(21) 式中, x_{\max} , x_{\min} 分别为 (X_k, Y_k) 中的最大值和最小值

部分代码如下:

maxA=max(A);


```

minA=min(A);
maxB=max(B);
minB=min(B);
for i=1:(nn-1)
    A(i)=(0.8/(maxA-minA))*(A(i)-minA)+0.1;
end

```

(3) 学习率的选取

学习率 $\alpha(t)$ 的正确选择对于保证 BP 网络收敛性而言非常重要，如果 $\alpha(t)$ 过大或过小，都会造成收敛速度和网络稳定性之间的矛盾。因此本文摒弃固定 $\alpha(t)$ ，采用基于黄金分割定律的变学习率方法：

$$\alpha(t+1) = \begin{cases} K_1 \alpha(t) & \lambda = 1 \\ 0.618 K_1 \alpha(t) & \lambda = -1 \end{cases} \quad (22)$$

$$\lambda = \text{sign}[G_{ij}^{(k-1)}(t) \bullet G_{ij}^{(k-1)}(t-1)] \quad (23)$$

(22) 式中， K_1 为学习率变换算子，取值范围要求 $1 < K_1 < 1.618$ 。

如式 (23) 可知，如果 $G_{ij}^{(k-1)}(t)$ 与 $G_{ij}^{(k-1)}(t-1)$ 方法相异，则 $\lambda = 1$ 表示 $\alpha(t)$ 下降过快，应减小步长；反之，若 $G_{ij}^{(k-1)}(t)$ 与 $G_{ij}^{(k-1)}(t-1)$ 方法相同，则 $\lambda = -1$ ，意味着方法下降过慢，应增加步长。

5.2.7 采用附加动量实现 BP 网络训练

运行训练程序，先弹出图 11 所示的菜单选择项，当选择 “W0=【-0.9】；B0=3；” 按钮时，弹出如图 所示的菜单选项及如图 12 所示的一维网络误差图。

部分代码如下

```

net=newcf(minmax(P),[1],{'logsig'});net.b{1}=B0;j=[-1:0.01:1];
for i=1:201
    net,iw{1,1}=j(i);
    y=sim(net,P);
    err=sumsq(y-T);
    error1=[error1 err];
end
figure(4);plot(j,error1);xlabel('输入向量');ylabel('一位误差曲线')
hold on;
Z2=menu('Use momentum constant of:',...
    '0.0',...
    '0.95[Default]');
if Z1==2
    [W0,dummy]=ginput(1);
end
D=H.*(1-H).*E;
gW=D*P';
dw=learnqdm([],[],[],[],[],[],gW,[],[],lp,ls);
ls.dw=dw;
TW=W+dw;
TA=logsig(TW*P+B*ones(1,400));
TE=T-TA;
TSSE=sumsq(TE);
if TSSE>SSE*err_ratio
    mc=0;
elseif TSSE<SSE
    mc=momentum;
end

```

```
figure(5);plot(w,error,'rp');
xlabel('输入向量');ylabel('随机权值变化曲线')
hold off;
figure(6);plot(error);
xlabel('输入向量');ylabel('训练误差曲线')
```

当采用附加变量法后 ($mc=0.95$)，网络的训练则可以自动的避免陷入这个局部极小值。训练结果如图 所示。网络的训练误差线落入局部极小值，在附加动量的作用下，继续向前产生一个正向斜率的运动，并跳入较浅的峰值，落入了全局最小值。然后，仍然在附加动量的作用下，达到一定的高度后（即产生一个 $SSE>SSE*1.04$ ）自然返回，并像子弹滚动一样来回左右摆动，知道停在最小值上。

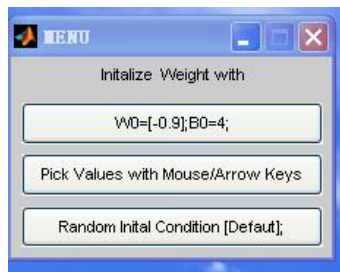


图 11 初始化权值菜单选择

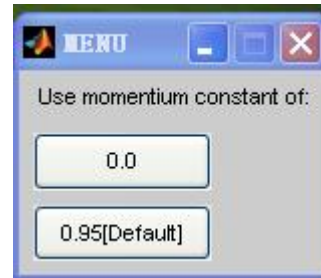


图 12 动量菜单选择

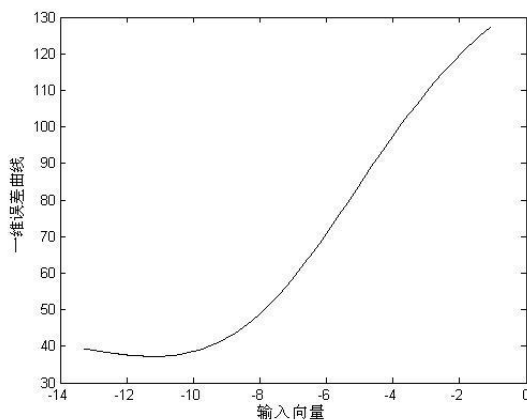


图 13 一位误差曲线

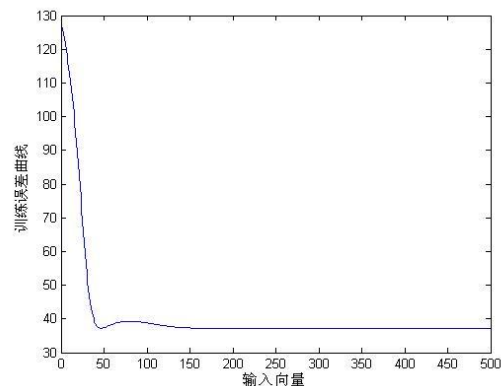


图 14 训练误差曲线

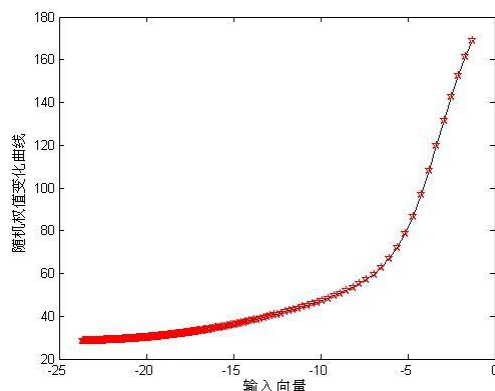


图 15 随机权值变化曲线

5.3 图像质量评价体系

对算法的评价一般有算法实现效果和算法计算时间两点，但是对于图像处理质量评价而言，仅仅依赖上述评价指标是不够的。图像质量评价标准依据评价主体不同可以分为主观和客观两类。

参赛队号#2785

1、主观评价：人为的观测待评测图像后，依据其视觉效果对图像进行评估，即对其进行打分，目前主要应用方法有两种：基于视觉特性评价和基于退化模型评价。

2、客观评价：通过计算处理后图像信息数据相关指标可以定量地评判图像质量，针对插值放大图像，更多地使用此方法。通过某种放大算法将小值图像放大到原始信息量后，可以依据某些判定指标或准则实现放大图像与原始图像信息比对，从而完成放大算法的质量评价。

目前客观评价主要采用的评价指标是均方误差（MSE）和峰值信噪比（PSNR），其表达式如式（24）（25），所示。

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N (I(i, j) - I'(i, j))^2}{M \times N} \quad (24)$$

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (I(i, j) - I'(i, j))^2} \quad (25)$$

其中 $I(i, j)$ 表示原始图像， $I'(i, j)$ 表示插值处理后图像， M 、 N 分别表示图像的长和宽。

均方误差反映放大图像与原图的逼近程度，均方误差越小表示放大图像越逼近原图，放大效果越好。平均梯度反映了图像中微小细节反差与纹理变化特征及清晰度，平均梯度越大表示放大图像清晰度越高，放大效果越好。

另外， MSE 和 $PSNR$ 被认为是一种行之有效的图像质量判定方法，但其作用原理均为求取像素平均误差后再进行相应计算评判，故此对于图像细节变化很难有较理想的反映。如果待比较的两幅图像总体像素平均差是相近的，依据上述指标定义，其图像放大质量较好，但事实上由于放大算法自身的缺陷和图像处理过程中不可避免地引入累积误差，两幅图像细节上不可能完全没有差别， MSE 和 $PSNR$ 判别图像放大质量具有一定的局限性。

因此我们采用另一种关注细节的客观指标反映图像放大质量，即下文介绍的平均梯度值指标。

平均梯度值 T 定义为：

$$T = \frac{1}{M \times N} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sqrt{\frac{(\Delta I_x^2 + \Delta I_y^2)}{2}} \quad (26)$$

$$\Delta I_x = I(i+1, j) - I(i, j)$$

$$\Delta I_y = I(i, j+1) - I(i, j)$$

综上所述，本文综合考虑人眼视觉效果主观定性评价和平均梯度值客观定量评价，首先利用主观评价分析算法优劣度，然后利用客观评价说明上述分析的正确性，二者结合，实现最终评测。

5.4 实验结果分析

形象比较和综合阐述本文提出的组合图像放大算法，分别通过最近邻插值、双线性插值、双立方插值、非线性小波插值和本文提出的内外组合插值法对图像作 10 倍放大处理，结果分别如图 17、图 18、图 19、图 20 和图 21 所示，主观评价结果见表 1，客观评价结果见表 2。



图 16 原始图像



图 17 最近邻插值 10 倍放大图

参赛队号#2785



图 18 双线性插值 10 倍放大图



图 19 双立方插值 10 倍放大图



图 20 小波插值放大 10 倍图



图 21 双线性+BP 神经网络

主观评价结果：从人眼的视觉效果上来看，图 17 最近邻插值的放大图像锯齿效应非常明显，这是由于最近邻插值的算法模糊匹配的机制决定的；图 18 双线性插值的放大图像相对平滑，但是由于平滑性太强，造成放大后图像边缘模糊，降低图像清晰度；图 19 双立方插值法图像较为清晰，较平滑，锯齿现象有所改善；图 20 小波插值法图像较清晰、平滑，分辨率有所改善；图 21 由本文提出的内部双线性插值和边缘神经网络插值相结合得到的放大图像，不仅内部效果平滑，无锯齿效应，而且边缘相对清晰连

贯，有较大的平滑性，既采用本文设计的基于图像信息分类组合插值放大算法对原始图像进行放大，可以消除最近邻插值的锯齿效应，改善双线性插值的边缘模糊现象。因此在主观评价方面，采用本文算法对图像放大是最优的。

表 1 图像主观评价表

插值类型	主观感受	图像轮廓	总体评价	处理耗时
最临近点插值 Nearest Neighbour	马赛克现象 严重	不清晰	差	0.014352 秒
双线性插值 Bilinear	图像较清晰， 不锐利	边缘不清晰， 有锯齿现象	折中	0.393390 秒
双立方插值 Bicubic	图像较清晰， 较平滑	锯齿现象有所 改善	较好	0.687144 秒
小波插值	图像较清晰， 平滑	分辨率有所改 善	较好	0.759762 秒
双线性+BP 神经 网络	图像清晰，平 滑	图像清晰，分 辨率高	好	0.772628 秒

客观评价结果：通过计算五种放大算法处理后的图像均方误差相差不大，但是本文提出的内部双线性插值与边缘神经网络相结合的插值放大算法平均梯度值要大于传统的线性插值算法，因此从客观评价的角度来说，采用本文算法对图像放大也是最优的。

表 2 图像客观评价表

算法	图像	峰值信噪比	均方误差	平均梯度值
最近邻插值	图 17	4.6	30.2254	7.83562
双线性插值	图 18	3.6	30.0095	8.34072
双立方插值	图 19	4.0	30.2213	8.14833
小波插值	图 20	3.8	30.1934	8.25308
双线性+神经网络	图 21	3.2	30.0003	8.32065
本文基于双线性内部插值与亚像素边缘检测神经网络最优组合，得到较好质量图像				

§ 6 模型的误差分析

6.1 图像噪点的误差：

噪点主要是指图像中不该出现的外来像素，通常由电子干扰产生。由于噪点在图像之中的广泛存在性，虽然我们已经使用软件对图片进行预处理处理噪点，但是还是会形成一些围绕在图像周围的无序点，对插值产生一定误差。

6.2 考虑图像放大内部插值的误差：

图像放大插值时，插值法在插值时，由于函数选择的不同，插值本身也存在着误差。

6.3 BP 神经网络的误差：

虽然 BP 神经网络得到广泛使用，但它也存在自身的限制与不足。在确定了 BP 网络的结构后，要通过输入和输出样本数据对网络进行训练，亦即对网络的阈值和权值进行

学习和修正,以使网络实现给定的输入/输出关系。由于阈值和权值选取的不同,训练产生的误差大小不同。另外 BP 算法可以使网络权值收敛到一个值解,但它不能保证所求的结果为误差超平面的全局最小解,有可能是一个局部极小解而造成误差。

§ 7 模型的评价

7.1 模型的优点:

1. 创新性的将图形分为内部和边缘两部分分别进行插值处理。内部运用现有成熟的线性插值法,图像的边缘运用 BP 神经网络算法进行插值。

2. 人工神经网络作为能够模拟自然人大脑神经系统结构功能特征的数学分析工具,不需要借助任何先验函数假设,通过自我解算即可把物理量之间复杂的因果关系通过适当的训练表现出来,并利用归纳总结的规律来预测未知。因此用神经网络模型对边缘插值是可行的。

3. 对比多种线性和非线性插值法,突出本文对图像放大处理的优势。

4. 对放大后图片质量进行定性和定量的评价,建立图片质量评价体系,完整直观的反映放大图片质量。

5. 利用 Malab、Excle 等软件对数据进行处理并作出各种图表,简便,直观,快捷,准确。

6. 本文巧妙运用流程图,将建模思路完整清晰的展现出来;

7.2 模型的缺点

1. 对于一些数据,我们在运算前对其进行了一些必要的处理,例如对图片进行降噪的预处理。

2. 模型中为使计算简便,使所得结果更理想化,忽略了算法复杂程度增加造成的运算时间延长。

§ 8 模型的改进与推广

8.1 模型改进

任意一种简单平面图形,经 matlab 得到的图像数据很大,而且在算法计算过程如双线性插值与神经网络模型中需要很多时间去处理由于数据太大造成计算机系统反应缓慢问题,严重影响到实时图像的处理速度。

多核计算机技术提升了多核处理器的资源利用率,可以有效缩短执行时间,发挥多核系统的优异性,对于大规模图像放大有很好的加速功能,提高效率。

8.2 模型的推广

以上已建的模型适用于简单二维色彩低分辨率图形。同样对于普遍用于各种场合中的三维图像适用,缩放处理也是很常用的。双线性插值的一个显然的三维空间延伸是三线性插值。

在二维模型的基础上,设 $z=f(x, y)$, 为简单起见,使用等距采样, X 轴方向和 Y 轴方向的区域等分数分别为 m 和 n, 故采样步长和各采样点为

$$xh = \frac{x_m - x_0}{m} \quad x_0, x_1 = x_0 + x_h, \dots, x_i = x_0 + i * x_h, \dots, x_m \quad (27)$$

$$yh = \frac{y_n - y_0}{n} \quad y_0, y_1 = y_0 + y_h, \dots, y_i = y_0 + i * y_h, \dots, y_n \quad (28)$$

采样数据为

参赛队号#2785

$$\begin{bmatrix} z_{00} = f(x_0, y_0) & z_{10} = f(x_1, y_0) & \dots & z_{m0} = f(x_m, y_0) \\ z_{01} = f(x_0, y_1) & z_{11} = f(x_1, y_1) & \dots & z_{m1} = f(x_m, y_1) \\ \dots & \dots & \dots & \dots \\ z_{0n} = f(x_0, y_n) & z_{1n} = f(x_1, y_n) & \dots & z_{mn} = f(x_m, y_n) \end{bmatrix}. \quad (29)$$

得到数据使用三线性插值模型进行三维图像的放大算法处理，使用三线性插值对三维坐标(x, y, z)进行插值得到放大图像。



图 22 三维原图

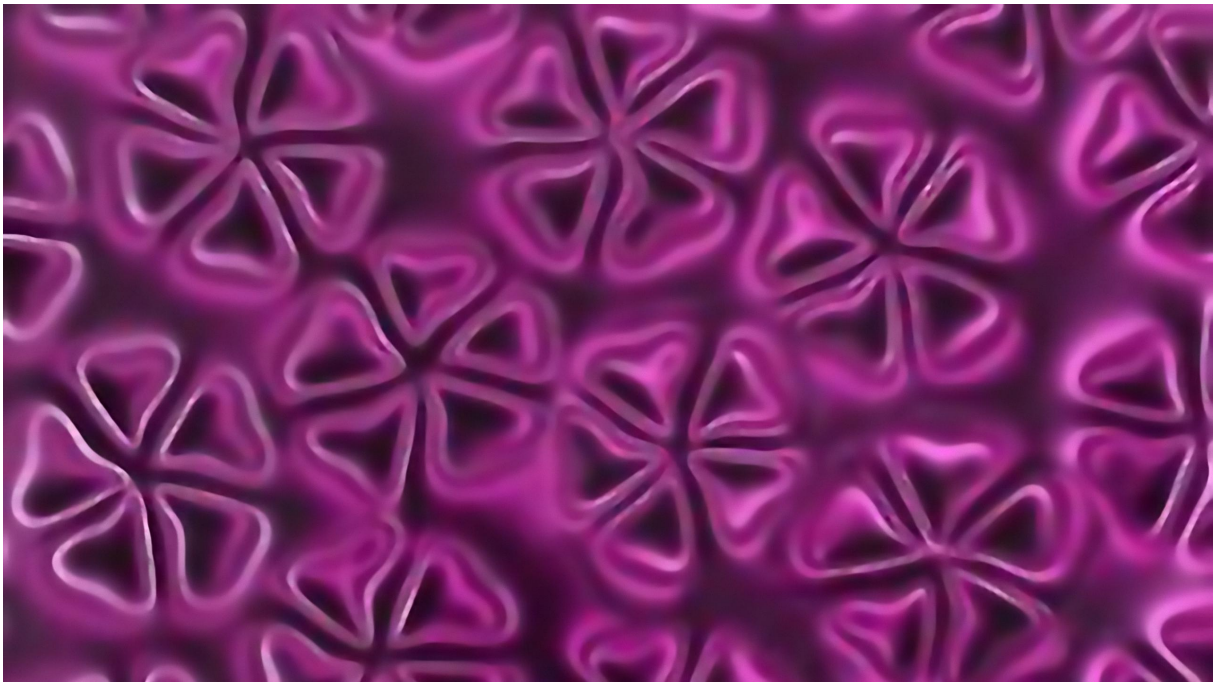


图 22 三维放大 10 倍图

§ 9 参考文献

- [1] Rafael C.Gonzalez, Richard E.Woods 主编，阮秋琦，阮宇智等译. 数字图像

参赛队号#2785

处理[M]. 电子工业出版社.

[2] 张德丰等主编. MATLAB 图像处理（第二版）[M]. 机械工业出版社.

[3] 田雨波等主编. 混合神经网络技术[M]. 科学出版社.

[4] 杨桂元, 黄己立主编. 数学建模[M]. 合肥: 中国科技大学出版社, 2008.

[5] 吴礼斌主编. 经济数学实验与建模[M]. 天津大学出版社. 2009.

附录

附录 1. 图像边缘横坐标归一化处理后结果

参赛队号#2785

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1								x						
2	0.10385	0.14231	0.18462	0.26923	0.31538	0.37692	0.41538	0.43846	0.46538	0.50385	0.57692	0.67308	0.76154	0.82308
3	0.10385	0.14231	0.19231	0.26538	0.32692	0.37692	0.41923	0.44231	0.46538	0.50385	0.57692	0.66923	0.76154	0.82308
4	0.1	0.14231	0.19615	0.27308	0.31923	0.37692	0.41538	0.44231	0.46923	0.50769	0.58846	0.66923	0.76154	0.82692
5	0.1	0.14231	0.19615	0.27308	0.32308	0.37308	0.41154	0.44231	0.46923	0.50385	0.58462	0.67308	0.75769	0.83077
6	0.1	0.14615	0.19231	0.27308	0.32692	0.37692	0.41538	0.44615	0.47308	0.50385	0.58846	0.67692	0.76154	0.83462
7	0.1	0.14231	0.19615	0.27692	0.32692	0.37692	0.41538	0.44231	0.47308	0.51154	0.59615	0.68077	0.76154	0.83462
8	0.1	0.15	0.19231	0.28077	0.32692	0.38077	0.41538	0.45	0.46923	0.50769	0.59231	0.68462	0.76923	0.83462
9	0.1	0.15385	0.20385	0.27692	0.32692	0.37692	0.41154	0.44615	0.46923	0.51154	0.59231	0.68462	0.76538	0.83462
10	0.10385	0.15	0.2	0.28077	0.33077	0.37692	0.41538	0.44231	0.46923	0.50769	0.59615	0.68462	0.76923	0.83846
11	0.1	0.15385	0.2	0.28077	0.33077	0.38077	0.41538	0.44615	0.47692	0.50769	0.59615	0.68077	0.76923	0.83462
12	0.1	0.15769	0.2	0.28077	0.33077	0.38077	0.41154	0.44615	0.47692	0.51538	0.59615	0.68462	0.76923	0.83846
13	0.1	0.15385	0.20385	0.28462	0.33462	0.38077	0.41923	0.44615	0.47692	0.50769	0.6	0.69231	0.76923	0.83846
14	0.1	0.15385	0.20385	0.28462	0.33462	0.38846	0.41923	0.44615	0.47692	0.51154	0.6	0.69231	0.77308	0.84231
15	0.10385	0.15385	0.20385	0.28077	0.33462	0.38462	0.41923	0.44615	0.47692	0.51154	0.6	0.69615	0.77308	0.84231
16	0.10385	0.15769	0.2	0.28462	0.33077	0.38462	0.42308	0.44615	0.47308	0.51154	0.60385	0.69615	0.77692	0.84615
17	0.10385	0.16154	0.20769	0.28462	0.33462	0.38462	0.41538	0.45	0.47692	0.51538	0.60385	0.69615	0.77692	0.85
18	0.10385	0.16538	0.21154	0.28846	0.33077	0.38846	0.41923	0.45	0.47692	0.51538	0.60769	0.7	0.77692	0.84615
19	0.10385	0.16538	0.20385	0.29231	0.33077	0.38462	0.41923	0.45	0.47308	0.51923	0.61154	0.7	0.78077	0.85385
20	0.10769	0.16538	0.20769	0.28846	0.33846	0.39231	0.41923	0.45	0.47692	0.51923	0.60769	0.70385	0.78077	0.85769
21	0.10769	0.16538	0.21154	0.28846	0.34231	0.38846	0.41923	0.45	0.47692	0.51923	0.60769	0.70769	0.78077	0.86154
22	0.10385	0.16538	0.21154	0.28846	0.33846	0.38846	0.42308	0.45	0.47692	0.52308	0.61923	0.70769	0.78077	0.86923
23	0.10769	0.16154	0.21154	0.28846	0.34231	0.39231	0.42308	0.45385	0.48462	0.52308	0.61538	0.70769	0.78077	0.87308
24	0.10769	0.16538	0.21154	0.29231	0.33846	0.39615	0.42308	0.45385	0.48077	0.52308	0.61154	0.71154	0.78077	0.88077
25	0.10385	0.16154	0.21538	0.29231	0.34615	0.39615	0.41923	0.45385	0.47692	0.52308	0.61154	0.71154	0.78077	0.89231
26	0.10769	0.16538	0.21538	0.29231	0.34615	0.39231	0.42308	0.45385	0.48077	0.52308	0.61923	0.71538	0.78462	0.89231
27	0.11154	0.16923	0.21538	0.29231	0.34615	0.39615	0.42692	0.45385	0.48462	0.52692	0.61538	0.71538	0.78462	0.9
28	0.11154	0.16538	0.21923	0.29615	0.34615	0.39615	0.42692	0.45385	0.48077	0.52692	0.62308	0.71923	0.78846	0.9
29	0.11154	0.16538	0.22308	0.29231	0.34615	0.39615	0.42692	0.45769	0.48077	0.52692	0.62692	0.71538	0.78846	0.9
30	0.11154	0.16923	0.22308	0.3	0.35	0.39615	0.43077	0.45385	0.48846	0.52692	0.62692	0.71154	0.79231	0.9
31	0.11154	0.16923	0.21923	0.29615	0.34615	0.39615	0.43077	0.45385	0.48077	0.53077	0.62692	0.71923	0.79231	
32	0.11154	0.16538	0.21923	0.30385	0.34615	0.39615	0.43077	0.45385	0.48077	0.52692	0.63077	0.71538	0.78846	
33	0.11923	0.16923	0.22692	0.3	0.35	0.39231	0.42692	0.46154	0.48462	0.53077	0.63077	0.72308	0.79615	
34	0.11923	0.16923	0.22692	0.29615	0.35	0.39231	0.43077	0.45769	0.48846	0.53462	0.63077	0.72308	0.8	
35	0.11923	0.17308	0.22692	0.30385	0.35	0.39615	0.43077	0.45769	0.49231	0.53462	0.62692	0.72308	0.79615	
36	0.12308	0.17308	0.22692	0.30769	0.35385	0.39615	0.43077	0.45769	0.48462	0.53462	0.63462	0.73077	0.79615	
37	0.12692	0.17308	0.22692	0.3	0.35	0.39615	0.43462	0.45769	0.48846	0.54231	0.63462	0.73077	0.8	
38	0.12308	0.17308	0.23077	0.30385	0.35385	0.40385	0.43462	0.45769	0.48846	0.54231	0.63462	0.73077	0.8	
39	0.12308	0.17308	0.23077	0.30385	0.36154	0.4	0.43846	0.45769	0.48846	0.54231	0.63846	0.73462	0.8	
40	0.12692	0.17308	0.23462	0.30385	0.36154	0.4	0.43846	0.46154	0.49231	0.54231	0.64231	0.74231	0.8	
41	0.12308	0.17308	0.23077	0.30769	0.36154	0.4	0.43462	0.46154	0.48846	0.54615	0.63846	0.73462	0.8	
42	0.11923	0.17308	0.23846	0.30769	0.36154	0.40385	0.43077	0.46538	0.49231	0.55	0.64231	0.74231	0.80385	
43	0.12308	0.17308	0.23462	0.31154	0.35769	0.40385	0.43462	0.45769	0.49615	0.55	0.64231	0.73846	0.80385	
44	0.11923	0.17692	0.23846	0.31154	0.35769	0.40769	0.43462	0.46154	0.49615	0.55	0.64615	0.74231	0.80769	
45	0.12308	0.17692	0.24231	0.30769	0.36154	0.40385	0.43462	0.46154	0.5	0.55385	0.64615	0.74615	0.80769	
46	0.12692	0.17692	0.23846	0.31154	0.36154	0.40769	0.43846	0.46154	0.49615	0.55385	0.65	0.74231	0.80769	
47	0.12692	0.18077	0.24231	0.31154	0.36154	0.40769	0.44231	0.46154	0.49231	0.55385	0.65385	0.74231	0.80769	
48	0.12692	0.18077	0.23846	0.31154	0.36154	0.40769	0.43846	0.46154	0.49231	0.55385	0.65	0.74615	0.81538	
49	0.12308	0.18462	0.24231	0.31538	0.36923	0.41154	0.43846	0.46538	0.49615	0.55769	0.65	0.74615	0.81154	
50	0.12692	0.17692	0.24615	0.31538	0.36154	0.40769	0.43846	0.46538	0.49615	0.55769	0.65385	0.75	0.81154	
51	0.12692	0.18077	0.25	0.31154	0.36923	0.40769	0.43846	0.46154	0.49615	0.55769	0.65385	0.75	0.81538	
52	0.12692	0.18077	0.24615	0.31154	0.36923	0.40769	0.43462	0.46923	0.5	0.56154	0.65769	0.75	0.81538	
53	0.13462	0.18462	0.25	0.31538	0.37308	0.40769	0.43846	0.46538	0.5	0.56154	0.65385	0.75	0.81154	
54	0.13462	0.18462	0.25	0.31923	0.36538	0.41154	0.43846	0.46154	0.5	0.56154	0.65769	0.75385	0.81154	
55	0.13077	0.18462	0.25769	0.31923	0.36923	0.41154	0.44231	0.46538	0.5	0.56538	0.66538	0.75385	0.81923	
56	0.13077	0.18846	0.25769	0.31538	0.37692	0.40769	0.44231	0.46923	0.5	0.56923	0.66538	0.75385	0.81538	
57	0.13462	0.18846	0.25769	0.31923	0.37308	0.41154	0.44231	0.46923	0.50385	0.56923	0.66923	0.75769	0.81538	
58	0.13846	0.18846	0.25769	0.32308	0.37308	0.41154	0.44231	0.46923	0.50385	0.56923	0.66538	0.75769	0.82308	
59	0.14231	0.19231	0.26538	0.31923	0.37308	0.41154	0.44231	0.46923	0.50385	0.57308	0.66154	0.75769	0.81923	
60	0.13846	0.18846	0.26154	0.32308	0.37308	0.41154	0.44231	0.46923	0.50769	0.57692	0.67308	0.76154	0.82308	

附录 2. 图像边缘纵坐标归一化处理后结果

参赛队号#2785

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2	0.18058	0.13741	0.10576	0.33022	0.3705	0.37626	0.29568	0.84621	0.81942	0.54892	0.18921	0.26115	0.37338	0.45971
3	0.21799	0.13741	0.1518	0.14892	0.10576	0.41367	0.4482	0.83957	0.26403	0.23525	0.23237	0.26978	0.45971	
4	0.48273	0.26691	0.2295	0.16043	0.16043	0.10863	0.52014	0.65923	0.14892	0.35612	0.31871	0.26115	0.34173	0.26691
5	0.48849	0.30144	0.73597	0.70144	0.3705	0.20935	0.52878	0.66574	0.19496	0.47122	0.51151	0.46835	0.43094	0.3964
6	0.53165	0.71295	0.15468	0.3705	0.63237	0.35612	0.55468	0.67226	0.40504	0.49712	0.56043	0.28129	0.48849	0.3964
7	0.54029	0.75612	0.31007	0.37914	0.10288	0.41655	0.64101	0.87877	0.43669	0.5518	0.18921	0.30432	0.24964	0.45396
8	0.55468	0.75324	0.17482	0.65827	0.14029	0.60647	0.11727	0.68528	0.46835	0.15755	0.51151	0.52014	0.32158	0.39065
9	0.55755	0.10576	0.38777	0.18345	0.19209	0.19784	0.13741	0.69179	0.47986	0.4741	0.20935	0.30432	0.34748	0.45108
10	0.56331	0.19496	0.53165	0.33309	0.21223	0.24388	0.33309	0.69831	0.56331	0.24388	0.31583	0.30719	0.36475	0.38777
11	0.78201	0.20647	0.73309	0.3705	0.1	0.36763	0.33885	0.70482	0.8482	0.25827	0.55755	0.46259	0.38777	0.44532
12	0.79928	0.26403	0.12014	0.57482	0.21223	0.41942	0.45108	0.71133	0.85971	0.47986	0.19209	0.51727	0.48561	0.37914
13	0.80216	0.30144	0.18345	0.69856	0.34748	0.43957	0.55755	0.71784	0.87122	0.50576	0.21799	0.24964	0.28705	0.43957
14	0.81367	0.71007	0.53453	0.13741	0.4741	0.10288	0.56906	0.72436	0.87986	0.59784	0.31295	0.31007	0.35036	0.43669
15	0.23813	0.19209	0.68993	0.59496	0.6295	0.18345	0.57482	0.73087	0.36475	0.2554	0.50288	0.22086	0.3705	0.37338
16	0.26691	0.71007	0.18058	0.18345	0.67554	0.27554	0.5777	0.73738	0.44245	0.5	0.50288	0.25252	0.33022	0.37338
17	0.29568	0.11439	0.31295	0.37914	0.16906	0.6036	0.1259	0.74389	0.46835	0.54604	0.55468	0.28417	0.42518	0.43381
18	0.60072	0.20072	0.38489	0.53741	0.20072	0.65252	0.34173	0.75041	0.48273	0.21799	0.19496	0.31583	0.33597	
19	0.60935	0.40791	0.54604	0.55468	0.38201	0.29281	0.35036	0.75692	0.56331	0.34748	0.19784	0.51151	0.36187	
20	0.62374	0.41367	0.73022	0.60647	0.47122	0.30719	0.35612	0.64388	0.88561	0.54317	0.23525	0.24676	0.48273	
21	0.66835	0.6036	0.13453	0.69568	0.67266	0.37914	0.45108	0.85827	0.89424	0.59496	0.5	0.51151	0.34173	
22	0.64388	0.61511	0.21511	0.17482	0.14029	0.42806	0.13165	0.66403	0.38201	0.16331	0.19784	0.22374	0.37914	
23	0.65827	0.75036	0.38489	0.33597	0.19784	0.43957	0.1518	0.88849	0.44532	0.24964	0.23525	0.31007	0.4223	
24	0.67266	0.10288	0.73022	0.50863	0.35612	0.12302	0.2036	0.89137	0.4741	0.59496	0.24388	0.45683	0.47986	
25	0.88849	0.25827	0.1777	0.61223	0.46835	0.1259	0.40504	0.9	0.5	0.24388	0.31007	0.32158	0.29856	
26	0.9	0.46259	0.2036	0.64964	0.22662	0.20647	0.65252	0.13165	0.1518	0.24964	0.5	0.45396	0.34748	
27	0.1777	0.58921	0.68417	0.69281	0.37914	0.21511	0.45396	0.44245	0.42518	0.54029	0.5518	0.2554	0.37338	
28	0.2036	0.59784	0.1	0.36187	0.46835	0.23525	0.65252	0.45683	0.44532	0.54029	0.26115	0.50576	0.37626	
29	0.69281	0.64676	0.11727	0.49712	0.62374	0.31007	0.66403	0.46547	0.47986	0.15468	0.20072	0.24964	0.41942	
30	0.69856	0.27842	0.72446	0.33885	0.12014	0.32446	0.67554	0.57194	0.50288	0.3446	0.26115	0.31007	0.30432	
31	0.71583	0.29281	0.13741	0.37914	0.18345	0.33597	0.68417	0.62086	0.60935	0.58633	0.49424	0.32734	0.47698	
32	0.2036	0.47122	0.68129	0.68993	0.46547	0.60072	0.69281	0.68417	0.40504	0.14892	0.26691	0.50288	0.30432	
33	0.26115	0.58921	0.69568	0.11727	0.62374	0.64964	0.20072	0.69568	0.4482	0.12878	0.30719	0.29856	0.36187	
34	0.76763	0.65252	0.31583	0.14317	0.66978	0.13453	0.40216	0.1259	0.60935	0.1777	0.49137	0.32158	0.41655	
35	0.7964	0.11439	0.37914	0.2036	0.1259	0.35324	0.43381	0.21511	0.15468	0.58345	0.54317	0.4482	0.4741	
36	0.12302	0.23525	0.72158	0.35899	0.11151	0.40504	0.45683	0.37338	0.20935	0.12302	0.24964	0.31871	0.2554	
37	0.14317	0.26403	0.67842	0.49424	0.19496	0.64676	0.69568	0.45683	0.36763	0.22374	0.30719	0.32446	0.30719	
38	0.23525	0.48561	0.69281	0.64388	0.22662	0.23813	0.70432	0.46835	0.39828	0.24964	0.48849	0.33885	0.36763	
39	0.26978	0.58058	0.14604	0.16619	0.37338	0.35899	0.71583	0.71871	0.44245	0.53165	0.54029	0.4482	0.27842	
40	0.29568	0.66115	0.32158	0.17194	0.61799	0.45396	0.1518	0.72446	0.45396	0.58058	0.19784	0.33022	0.31007	
41	0.72158	0.66978	0.67554	0.49137	0.66691	0.59784	0.16906	0.73309	0.51439	0.11439	0.24388	0.22662	0.3705	
42	0.83094	0.70144	0.71871	0.64101	0.10576	0.1	0.20072	0.12014	0.52302	0.21511	0.48561	0.24101	0.37626	
43	0.83957	0.7446	0.71583	0.68705	0.18345	0.2036	0.36763	0.14604	0.55468	0.33885	0.53741	0.29281	0.41367	
44	0.84532	0.11727	0.1777	0.10576	0.37338	0.24101	0.39065	0.37338	0.26115	0.53165	0.30144	0.33309	0.47122	
45	0.85683	0.5777	0.32158	0.1518	0.40504	0.25827	0.4741	0.45971	0.3964	0.58058	0.20072	0.3446	0.31583	
46	0.86835	0.67554	0.67266	0.21511	0.45683	0.41367	0.48273	0.46835	0.45396	0.33597	0.22086	0.44245	0.2554	
47	0.39065	0.19496	0.68705	0.35899	0.66403	0.45971	0.73022	0.74173	0.48849	0.52878	0.22086	0.28705	0.31871	
48	0.72158	0.56619	0.1259	0.68417	0.1	0.46547	0.7446	0.75324	0.5518	0.5777	0.48273	0.49424	0.33885	
49	0.76187	0.68129	0.18633	0.14317	0.21223	0.64388	0.75612	0.76763	0.21223	0.24388	0.53453	0.24101	0.3446	
50	0.88273	0.69568	0.67266	0.18345	0.45396	0.10576	0.19496	0.77626	0.24964	0.52302	0.2554	0.26978	0.32158	
51	0.89712	0.74173	0.32446	0.34173	0.61223	0.26403	0.36475	0.10576	0.3705	0.1777	0.29856	0.33885	0.40791	
52	0.16331	0.30719	0.68705	0.35899	0.66115	0.36187	0.50288	0.2036	0.38777	0.33022	0.53165	0.35612	0.32446	
53	0.20072	0.39065	0.22086	0.48561	0.11727	0.47986	0.51151	0.40216	0.39353	0.52014	0.20072	0.43669	0.40504	
54	0.25827	0.52302	0.66691	0.63813	0.1259	0.48561	0.52302	0.47122	0.46259	0.51727	0.52878	0.49424	0.46835	
55	0.26978	0.56043	0.15755	0.1	0.19784	0.49137	0.52878	0.61511	0.49712	0.56906	0.47698	0.29856	0.26978	
56	0.29856	0.73885	0.32734	0.13741	0.41079	0.50288	0.76475	0.78777	0.52302	0.32446	0.22662	0.30719	0.27266	
57	0.71871	0.1	0.70719	0.2036	0.4482	0.10863	0.77914	0.79353	0.60072	0.18345	0.25827	0.3446	0.32734	
58	0.39353	0.14317	0.20935	0.63813	0.60935	0.14892	0.78777	0.1	0.15755	0.51439	0.47698	0.43669	0.27554	
59	0.71583	0.1777	0.19784	0.68129	0.66115	0.2036	0.79353	0.41367	0.21223	0.56619	0.20647	0.23813	0.33885	
60	0.75899	0.18921	0.67554	0.18633	0.11151	0.27842	0.63318	0.43094	0.26403	0.22086	0.28993	0.26115	0.46259	
61		0.69568	0.70432	0.35612	0.13165	0.29281	0.63969	0.46547	0.46259	0.56331	0.29568	0.29568	0.39928	