
2015 高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则。

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，我们将受到严肃处理。

我们参赛选择的题号是（从 A/B/C/D 中选择一项填写）：_____

我们的参赛报名号为（如果赛区设置报名号的话）：_____

所属学校（请填写完整的全名）： 重庆邮电大学

参赛队员（打印并签名）： 1. 常兴甲

2. 刘孟鑫

3. 魏佩儒

指导教师或指导教师组负责人（打印并签名）：_____

日期： 2016年8月9日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2015 高教社杯全国大学生数学建模竞赛

编号专用页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

评阅人										
评分										
备注										

全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

基于 MATLAB 的相似截图与图形匹配问题的研究

摘要

本文主要研究相似性最大截图以及截图识别匹配问题，建立了基于 MATLAB 的图片数据简化与图片匹配模型，并且通过计算截图所对应矩阵相关系数来判断图片是否相似。

对于问题一，我们通过使用 MATLAB 将图片进行二值化，然后用 MATLAB 将其进行聚类，分为四类 (A_1, A_2, A_3, A_4)。将每一类图片进行类内重合，阈值为 0.2 即相应部分目标图像出现概率大于 0.2 保留图像，其中 A_4 中图形完全一样，从而验证了分类的合理性。得到四张类内重合图之后，经观察可以进一步简化，将 A_1 与 A_2 ， A_3 与 A_4 的类内重合图重合之后，得到代表图 S_1 和 S_2 。将图片简化之后，若采用穷举法计算相应截图的相似度是不可取的（数据太大不易于操作），此时我们将图片进行分割计算分割后图片的相似度，找到相似度最大的区域，并将其扩大，也就是题目中所要求截图的范围： $(40*40)$: $S_1 [256, 218]$, $S_2 [46, 218]$ 相似度: 0.2309; $(20*20)$: $S_1 [41, 267]$, $S_2 [146, 294]$ 相似度: 0.2506。最后在截图所在范围内采用穷举法进行相关系数的计算，得到最后结果： $(40*40)$ $S_1 [246, 208]$ 、 $S_2 [36, 208]$ 相似度: 0.3328; $(20*20)$ $S_1 [37, 279]$ 、 $S_2 [153, 305]$ 相似度: 0.3067。

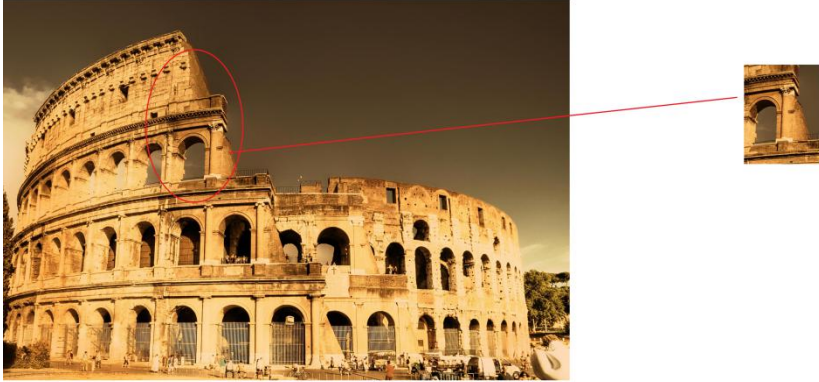
对于问题二，我们需要得到截图位置以及原图标号。我们采用与问题一中相同的方法将图片进行简化，再确定附件中截图原图所属类别以及截图位置：（截图一） $A_4 [115, 340]$ 相似度为 0.5707；（截图二） $A_3 [237, 168]$ 相似度为 0.4578。然后将原图所属类别中图片的截图位置进行截图，得到的图片与附件中截图进行相似度计算，得到最大值的图片就是附件中截图的原图：得到截图一的原图为 A_4 中任意一张，截图二的原图为标号为 tuxing1 的图片。最后用 MATLAB 将 60 张图片进行验证，得到结果与模型结果相符合，可见模型二在解决问题的同时既是合理的又大大降低了运算的复杂度。

本文最大的特点就是将大量数据进行合理简化，再根据简化后的结果采用合适的方法解决问题，这样既保证了解决问题的合理性，又提高了效率。

关键词 模糊聚类分析 穷举法 相关系数法 图片匹配 简化算法

一、问题重述

近期，江苏卫视正在热播节目“最强大脑”。曾经有这样一期节目，选手先观察近百幅人类历史上出现过的知名建筑图片（沙画，每幅大约 1 平米），然后由考官随机地从图片中选取一副 4cm X 4cm, 2cm X 2cm 等规格的截图，比如：
（下图来自于网络）

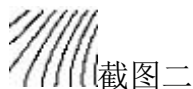


再让选手只观察截图（上图右）判别截图出自于哪一幅沙画。节目中选手几乎是以不可想像的速度做出了正确的选择。看完节目，觉得考官的选择还不够好，应该选择这些图片中相似性最大的部分，而不能选择那些有显著特点的截图，那么题目的难度更会显著增加。

解决以下问题：

1. 如何选择出那些相似性比较高的截图。并对附件中的图片选择出 40X40 像素及 20X20 像素的相似性最高的截图（截图仅限出现在附件图片中的矩形框内，空白截图除外）。

2. 建立模型解决对于给定的截图出自于第几幅图片的何处位置的问题。



二、模型假设

1. 假设所截图片与 x, y 轴分别平行（垂直）。
2. 假设不考虑图片的旋转、反转相似。
3. 假设图片中坐标轴不影响计算结果。
4. 假设未经特定说明，文中给出的都是像素坐标。
5. 文中截图区域都用其左上角坐标表示。

三、定义与符号说明

L_k	第 k 张图片。
A_j	类型为 j 的图片集。
C_j	A_j 的重合图。

S_i	C_1, C_2 的重合图为 S_1 ; C_3, C_4 的重合图为 S_2 。
-------	---

四、问题分析

1. 问题一的分析

问题一要求我们找出 60 张图片中相似性最高的截图，也就是计算图片中部分截图（40*40 与 20*20）与其他图片部分截图的相似度，并且找出最大值。我们采用将图片转化为矩阵，建立模型，用相关系数法计算部分截图的相似度的方法进行计算，并找出最大值（最佳截图位置），即题目中所求部分。

2. 问题二的分析

问题二要求我们找出附件中截图源自哪张图片，即在哪个原图中可以找出与截图相似度最大部分。根据问题一中所得到的结果，我们采取穷举法，用 MATLAB 相关系数检验函数计算截图矩阵与各原图矩阵每一部分的相似度，并找出最大值，取得最大值的位置即题中所需要的结果。

五、模型的建立与求解

1.1 问题一模型的建立

1) 聚类分析法^[1]

数据分类中，常用的分类方法有多元统计中的系统聚类法、模糊聚类分析等。在模糊聚类分析中，首先要计算模糊相似矩阵，而不同的模糊相似矩阵会产生不同的分类结果；即使采用相同的模糊相似矩阵，不同的阈值也会产生不同的分类结果。该模型通过 MATLAB 将 60 张图片的模糊相似矩阵求解，并确定阈值为 0.2，最终得到四类图片。

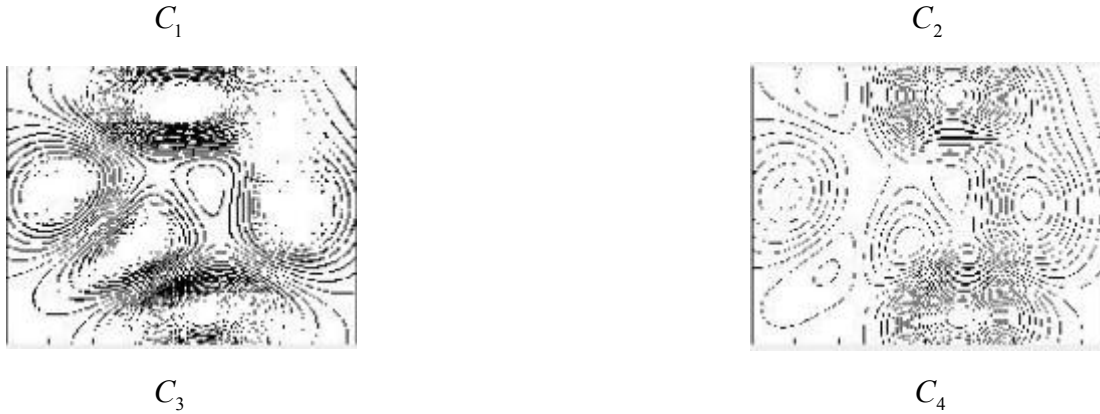
2) 图片数据的简化

因为题目中给出图片过多，为了简化算法并且减少运算量，我们将图片分类最后得到两张代表图，这样做大大减少了运算量，并且经过验证之后发现是合理的。首先我们将六十张图片构建成六十个模糊矩阵，采用聚合分析法将其分为四类：

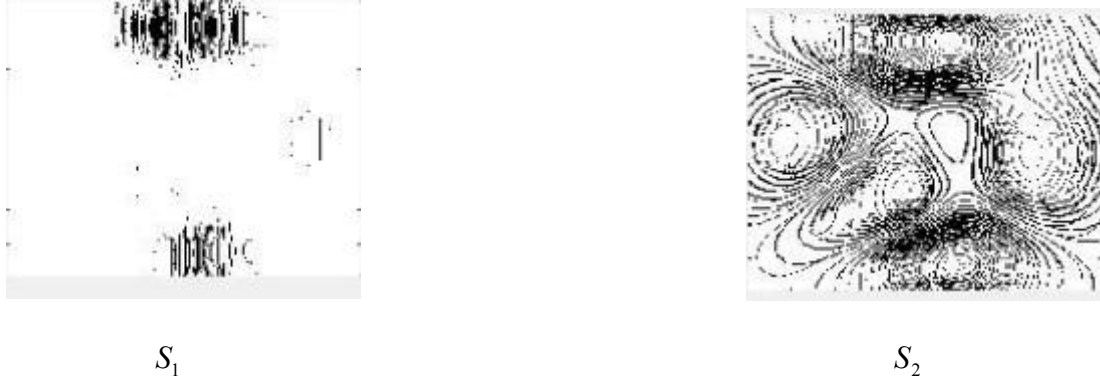
A_1	$L_k (k = 40, 41, 43, 44, 46, 47, 49, 50)$
A_2	$L_k (k = 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26, 28, 29, 31, 32, 34, 35, 37, 38)$
A_3	$L_k (k = 1, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60)$
A_4	$L_k (k = 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48)$

这四类图片中每一类图片类内相似度很高，所以我们将每一类图片进行类内重合，保留目标图像（黑色线条）出现较多的区域得到的结果如下：





我们发现类别四中图片完全重合，这又验证了分类的合理性。
 经观察可知， C_1 与 C_2 目标图像分布情况相近， C_3 与 C_4 目标图像分布情况相似。所以我们进一步简化数据，将 C_1 与 C_2 ， C_3 与 C_4 进行重合，保留目标图像出现最多的区域，最后得到两张图片 S_1, S_2 ，分别表示 $C_1 C_2$ ， $C_3 C_4$ ：



经过一系列的处理，我们最终把六十张图片简化为两张图片（ S_1, S_2 ），用MATLAB^[2]通过相关系数法检验，得到代表图与原图相似度很高，所以这样简化是合理的。因此在六十张图片中找相似性最高的截图就转化为在两张图片中找相似性最高的截图，这样大大减少了计算数据，并且合理。

1. 2. 问题一的求解

在得到两幅代表图之后，我们采用将图片分割求出相似度最大的区域（大截图），并且将大截图扩大，然后用穷举法验证大截图中截图的相似度，从而找到最大值。

1) 截图为 40*40 时

首先将两幅图片分割为 8*8 个矩形（44*54），并用相关系数法求出两幅代表图个大截图之间的相似度，找出相似度最大的大截图，也就是截图所落在的区域。经过 MATLAB 编程计算可得：

	位置	相似度
S_1	[256, 218]	0. 2309
S_2	[46, 218]	

为了减小误差，将大截图扩大，上下左右都加 10，此时大截图在 S_1 中的坐标为：[246, 208]；在 S_2 中的坐标为：[36, 208]。题目中所求截图就在大截图中。

用穷举法计算两幅代表图中大截图内所含截图的相似度，并且找出最大值，此时最大值所对应的截图即为题目中所求截图，经 MATLAB 计算结果如下表所示：

	位置	相似度
S_1	[255, 216]	0.3328
S_2	[42, 222]	

2) 截图为 20*20 时

首先将两幅图片分割为 16*16 个矩形 (22*27)，并用相关系数法求出两幅代表图个大截图之间的相似度，找出相似度最大的大截图，也就是截图所落在的区域。经过 MATLAB 编程计算可得：

	位置	相似度
S_1	[41, 267]	0.2506
S_2	[146, 294]	

为了减小误差，将大截图扩大，上下左右都加 5，此时大截图在 S_1 中的坐标为：[36, 262]；在 S_2 中的坐标为：[141, 289]。题目中所求截图就在大截图中。用穷举法计算两幅代表图中大截图内所含截图的相似度，并且找出最大值，此时最大值所对应的截图即为题目中所求截图，经 MATLAB 计算结果如下表所示：

	位置	相似度
S_1	[37, 279]	0.3067
S_2	[153, 305]	

2.1 问题二模型的建立

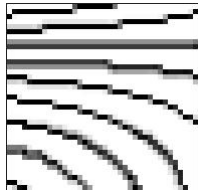
1) 图形的简化

在模型一的基础上，将 60 张图片分为 4 类，于是题目就简化成在四张图片中找出截图的位置与原图，这样大大简化了计算。

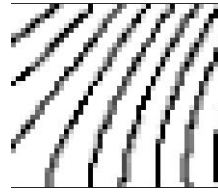
2) 截图的处理

通过观察截图与对比原图，发现截图有部分空白分布不合理，因此将截图不合理部分截去，此时截图大小为：

截图一	37*38
截图二	38*44



截图一



截图二

3) 截图区域的确定

将截图与四张代表图通过 MATLAB 转化为灰度矩阵，然后用穷举法计算截图与代表图的相关系数，找出最大值，则可确定原图的类别以及在原图中的位置。

4) 原图的确定

确定类别之后，计算该类图片所对应位置与截图的相似度，找出相似度最大的图片，即截图的原图。

2.2 问题二的求解

1) 截图一原图与位置的确定

将截图一与四类图片代表图通过 MATLAB 进行穷举对比，计算其相关系数，得到最大相似系数为：0.5707，最大截图类别为类型四，截图位置为：[115, 340]。又因为第四类图片完全一致，则截图一的原图可能是 A_4 ($L_k (k = 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48)$) 中任意一张图片。



截图一



L_3 截图

2) 截图二原图与位置的确定

将截图二与四类图片代表图通过 MATLAB 进行穷举对比，计算其相关系数，得到最大相似系数为：0.4578，最大截图类别为 A_3 ，截图位置为：[237, 168]。

将类别三中所有图片在截图位置截图，依次计算与截图二的相关系数，得到最大值所在图片为 L_1 。



截图二



L_1 截图

2.3 问题二的检验

采用穷举法计算 60 张图片与附件中截图的相似度，通过数学软件 MATLAB 计算了一个小时才计算完毕，并且得到的结果与文中计算的结果一致。

六、模型评价与推广

1. 模型一的评价与推广

我们首先将图片转化为模糊矩阵，并通过聚合分析法将其分为四类，这样做的目的是简化计算，减少数据的复杂度。然后经观察发现类别一与类别二、类别三与类别四目标图形分布相近，因此将图片简化为两张代表图，这样做不仅大大降低了计算的复杂度，并且经过验证是合理的。

对于模型一，其优点是大大简化了运算的复杂度且误差较小，这种简化可以提高运算的效率，为实际运用时打下了基础。将图形相似转化为纯粹的数学问题，并且用数学软件进行求解，这样做不仅简化运算时候的成本，而且提高了效率、精度。缺点是对类型不相似的图片分类会比较复杂。

在大数据时代，我们更加在意的是数据之间的关系，将杂乱无章的数据之间的联系找出来，从而简化数据。对于模型一，我们可应用于图像分析之前的预处理，以及对图片分类，图片溯源等方面。

2. 模型二的评价与推广

我们首先将图片的相似度转化为数学中矩阵的相似度，即计算图片所对应二值化矩阵之间的相关系数，从而将复杂的图像匹配问题转化为简单的数学问题，并且转化后的数学问题可以很容易的通过 MATLAB 进行求解。

对于模型二，其优点是将复杂的图像匹配，以及图片相似度计算等问题转化为简单的数学问题，这种转换降低了解题的难度。缺点是对于比较复杂的图像计算量会大大增加，那种情况下我们可以通过感知哈希算法来解决问题。

该模型适用于简单图片匹配，图片溯源等一些图片不复杂的问题。

参考文献

- [1]. 姜启源，谢金星，叶俊，数学模型（第三版）[M]，北京：高等教育出版社，2003 年。
- [2]. 周晓阳，数学实验与 Matlab[M]，武汉：华中科技大学出版社，2002 年；

附录：

1. 聚合分析代码

```
clear;clc;
ttmp01 = zeros(16,16);
line01 = zeros(60,256);
for i = 1:60 %将原图处理为 60 张含 16*16 矩阵的元胞数组 tmp01
    ii=int2str(i);
    x=imread(['D:\MATLAB\workspace\pic\picture\down',ii,'.bmp']);
    %读取当前文件夹, ' ' 可以变成其他文件

    xtmp = 3;
    ytmp = 1;
    for ii = 1:16 %将一幅图分为 16*16 的元胞数组矩阵
        for jj = 1:16
            tmp{ii,jj} = x(xtmp:xtmp+20, ytmp:ytmp+26);
            ytmp=ytmp+27;
        end
        xtmp=xtmp+21;
        ytmp = 1;
    end

    % areatmp = x(4:338,2:433);
    area = sum(sum(x));
    % area = 149205 - area;
    areaavr = area/256;

    for ii = 1:16
        for jj = 1:16
            xt = tmp{ii,jj};
            xtaarea = sum(sum(xt));
            % xtaarea = 567 - xtaarea;
            if xtaarea < areaavr
                ttmp01(ii,jj) = 0;
            else
                ttmp01(ii, jj) = 1;
            end
        end
    end

    end
    tmp01{i} = ttmp01(1:16,1:16);
end
```

```
subplot(221);imshow(tmp01{1});
subplot(222);imshow(tmp01{2});
subplot(223);imshow(tmp01{3});
subplot(224);imshow(tmp01{4});
```

```
%聚类分析分析过程
```

```
iii=1;
for i = 1:60          %将元胞数组分为 60*256 的数组，用于聚类分析
    ttmp = reshape(tmp01{i},16,16);
    for ii = 1:16
        for jj = 1:16
            line01(i, iii) = ttmp(ii,jj);
            iii=iii+1;
        end
    end
    end
    iii=1;
end
T=clusterdata(line01,4);
```

2. 图片重合程序:

```
%画重合图
sx = find(T == 1);      %修改 T == X, 即可改变类
clear juleicell
clear juleinum
clear juleitmp
for i = 1:length(sx)
    ii=int2str(sx(i));
    juleicell{i} =
imread(['D:\MATLAB\workspace\pic\julei\down',ii,'.bmp']);    %读取当前文件夹, ' '可以变成其他文件
end
```

```
for ii=1:length(sx)
    juleitmp = juleicell{ii};
    for iii=1:343
        for jjj = 1:435;
            if(juleitmp(iii, jjj) == 1) juleitmp(iii, jjj)=0;
            else juleitmp(iii, jjj)=1;
            end
        end
    end
end
```

```

    end
    juleicell{ii} = juleitmp;

end

%画图 percent 为百分比
percent = 0.2;
percenttmp = percent*length(sx);
juleinum = zeros(343,435);

for ii = 1:length(sx)
    juleinum = juleinum + juleicell{ii};
end

for iii=1:343
    for jjj = 1:435;
        if(juleinum(iii, jjj) >= percenttmp) juleinum(iii, jjj) = 0;
        else juleinum(iii, jjj) = 1;
        end
    end
end

end
subplot(221);imshow(juleinum);

juleinum1 = juleinum;          %%改变, juleinum1-4

%画重合图
sx = find(T == 2);          %修改 T == X, 即可改变类
clear juleicell
clear juleinum
clear juleitmp
for i = 1:length(sx)
    ii=int2str(sx(i));
    juleicell{i} =
imread(['D:\MATLAB\workspace\pic\julei\down', ii, '.bmp']);    %读取当前文件夹, ' '可以变成其他文件
end

for ii=1:length(sx)
    juleitmp = juleicell{ii};
    for iii=1:343

```

```

        for jjj = 1:435;
            if(juleitmp(iii, jjj) == 1) juleitmp(iii, jjj)=0;
            else juleitmp(iii, jjj)=1;
            end
        end
    end
    juleicell{ii} = juleitmp;

end

%画图 percent 为百分比
percent = 0.2;
percenttmp = percent*length(sx);
juleinum = zeros(343, 435);

for ii = 1:length(sx)
    juleinum = juleinum + juleicell{ii};
end

for iii=1:343
    for jjj = 1:435;
        if(juleinum(iii, jjj) >= percenttmp) juleinum(iii, jjj) = 0;
        else juleinum(iii, jjj) = 1;
        end
    end
end

end
subplot(221);imshow(juleinum);

juleinum2 = juleinum;                %%改变, juleinum1-4

%画重合图
sx = find(T == 3);                %修改 T == X, 即可改变类
clear juleicell
clear juleinum
clear juleitmp
for i = 1:length(sx)
    ii=int2str(sx(i));
    juleicell{i} =
    imread(['D:\MATLAB\workspace\pic\julei\down', ii, '.bmp']);    %读取当
    前文件夹, ' '可以变成其他文件
end

```

```

for ii=1:length(sx)
    juleitmp = juleicell{ii};
    for iii=1:343
        for jjj = 1:435;
            if(juleitmp(iii, jjj) == 1) juleitmp(iii, jjj)=0;
            else juleitmp(iii, jjj)=1;
            end
        end
    end
    juleicell{ii} = juleitmp;

end

%画图 percent 为百分比
percent = 0.2;
percenttmp = percent*length(sx);
juleinum = zeros(343, 435);

for ii = 1:length(sx)
    juleinum = juleinum + juleicell{ii};
end

for iii=1:343
    for jjj = 1:435;
        if(juleinum(iii, jjj) >= percenttmp) juleinum(iii, jjj) = 0;
        else juleinum(iii, jjj) = 1;
        end
    end
end
end
subplot(221);imshow(juleinum);

juleinum3 = juleinum;          %%改变, juleinum1-4

%画重合图
sx = find(T == 4);          %修改 T == X, 即可改变类
clear juleicell
clear juleinum
clear juleitmp
for i = 1:length(sx)
    ii=int2str(sx(i));

```

```

    juleicell{i} =
    imread(['D:\MATLAB\workspace\pic\julei\down', ii, '.bmp']);    %读取当前文件夹, ' '可以变成其他文件
end

for ii=1:length(sx)
    juleitmp = juleicell{ii};
    for iii=1:343
        for jjj = 1:435;
            if(juleitmp(iii, jjj) == 1) juleitmp(iii, jjj)=0;
            else juleitmp(iii, jjj)=1;
            end
        end
    end
    juleicell{ii} = juleitmp;

end

%画图 percent 为百分比
percent = 0.2;
percenttmp = percent*length(sx);
juleinum = zeros(343, 435);

for ii = 1:length(sx)
    juleinum = juleinum + juleicell{ii};
end

for iii=1:343
    for jjj = 1:435;
        if(juleinum(iii, jjj) >= percenttmp) juleinum(iii, jjj) = 0;
        else juleinum(iii, jjj) = 1;
        end
    end
end
end
subplot(221);imshow(juleinum);

juleinum4 = juleinum;    %%改变, juleinum1-4

for iii=1:343
    for jjj = 1:435;

```

```

        if(juleinum1(iii, jjj) == 0 || juleinum2(iii, jjj) == 0)
juleinumx1(iii, jjj) = 0;
        else juleinumx1(iii, jjj) = 1;
        end
    end
end

for iii=1:343
    for jjj = 1:435;
        if(juleinum3(iii, jjj) == 0 || juleinum4(iii, jjj) == 0)
juleinumx2(iii, jjj) = 0;
        else juleinumx2(iii, jjj) = 1;
        end
    end
end
End
subplot(221);imshow(juleinumx1);
subplot(222);imshow(juleinumx2);

```

3. 相关系数计算函数：

```

    xtmp = 4;
    ytmp = 2;
for ii = 1:16          %将一幅图分为 16*16 的元胞数组矩阵
    for jj = 1:16
        juleinumcell1{ii, jj} = juleinumx1(xtmp:xtmp+20, ytmp:ytmp+26);
        ytmp=ytmp+27;
    end
    xtmp=xtmp+21;
    ytmp = 2;
end

xtmp = 4;
ytmp = 2;
for ii = 1:16          %将一幅图分为 16*16 的元胞数组矩阵
    for jj = 1:16
        juleinumcell2{ii, jj} = juleinumx2(xtmp:xtmp+20, ytmp:ytmp+26);
        ytmp=ytmp+27;
    end
    xtmp=xtmp+21;
    ytmp = 2;
end

subplot(221);imshow(juleinumcell1{1, 1});
subplot(222);imshow(juleinumcell2{1, 1});

```

```

juleitmp = zeros(21,27);
clear juleiline1
ix=1;
  for ii = 1:16
    for jj = 1:16
      juleitmp = juleinumcell1{ii,jj}
      juleitmp = reshape(juleitmp,1,567);
      juleiline1{ix} = juleitmp;
      ix=ix+1;
    end
  end
end

juleitmp = zeros(21,27);
clear juleiline2
ix=1;
  for ii = 1:16
    for jj = 1:16
      juleitmp = juleinumcell2{ii,jj}
      juleitmp = reshape(juleitmp,1,567);
      juleiline2{ix} = juleitmp;
      ix=ix+1;
    end
  end
end

corrlab = zeros(256,256);
for iii = 1:256
  for jjj = 1:256
    caltmp1 = sum (juleiline1{iii},2);
    caltmp2 = sum (juleiline2{jjj},2);
    if(caltmp1 < 453)
      if(caltmp2 < 453)
        C=abs(corr2(juleiline1{iii}, juleiline2{jjj}));
        corrlab(iii, jjj) = C;
      else
        corrlab(iii, jjj) = 0;
      end
    else
      corrlab(iii, jjj) = 0;
    end
  end
end
end
end

```

```
[x y]=find(corrcoef(juleinumcell2{1,2},juleinumcell2{1,3})) %找最大
C=corrcoef(juleinumcell2{1,2},juleinumcell2{1,3});
subplot(221);imshow(juleiline1{43});
subplot(223);imshow(juleiline2{124});
```

```
pxn1 = juleinumx1(41:41+30,267:267+36);
pxn2 = juleinumx2(146:146+30,294:294+36);
subplot(221);imshow(pxn1);
subplot(222);imshow(pxn2);
subplot(223);imshow(juleiline1{43});
subplot(224);imshow(juleiline2{124});
```

```
clear caltmp1;
clear caltmp2;
c = zeros(12,18);
```

```
ix=1;
iy=1;
for iii = 1:12
    for jjj = 1:18
        caltmp1 = pxn1(iii:iii+39, jjj:jjj+39);
        for kkk = 1:12
            for sss = 1:18
                caltmp2 = pxn2(kkk:kkk+39, sss:sss+39);
                c(ix, iy)=abs(corr2(caltmp1, caltmp2));
                iy=iy+1;
            end
        end
        ix=ix+1;
        iy=1;
    end
end
end
```

```
[x y]=find(c==max(max(c))) %找最大
```

```
px1=pxn1(1:1+19,17:17+19);
px2=pxn2(12:12+19,16:16+19);
subplot(221);imshow(px1);
subplot(222);imshow(px2);
```