

MATLAB 手写汉字识别

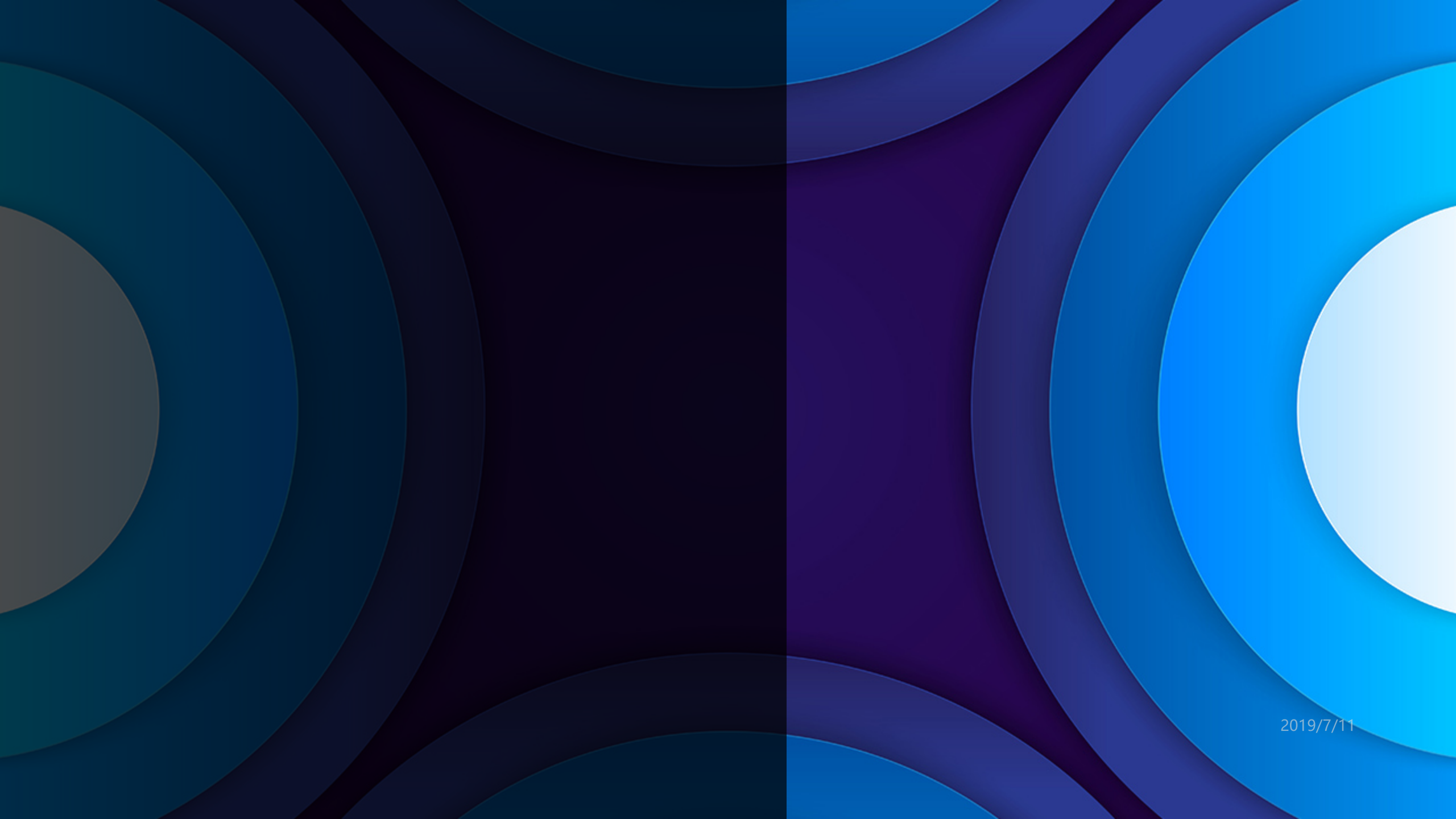
—— 徐震



项目摘要

项目摘要

本项目通过**图像分块**（包括**扇形分块**和**矩形分块**以及**直方图**）的方法对手写汉字图像进行特征提取和相似度计算。相似度采用开发者调整的**巴氏距离**计算，并对不同分块方式得到的相似度进行整合。根据相似度确定被测图片属于哪个汉字，输出与该图片最相似的十张图片。



2019/7/11

目录

目录

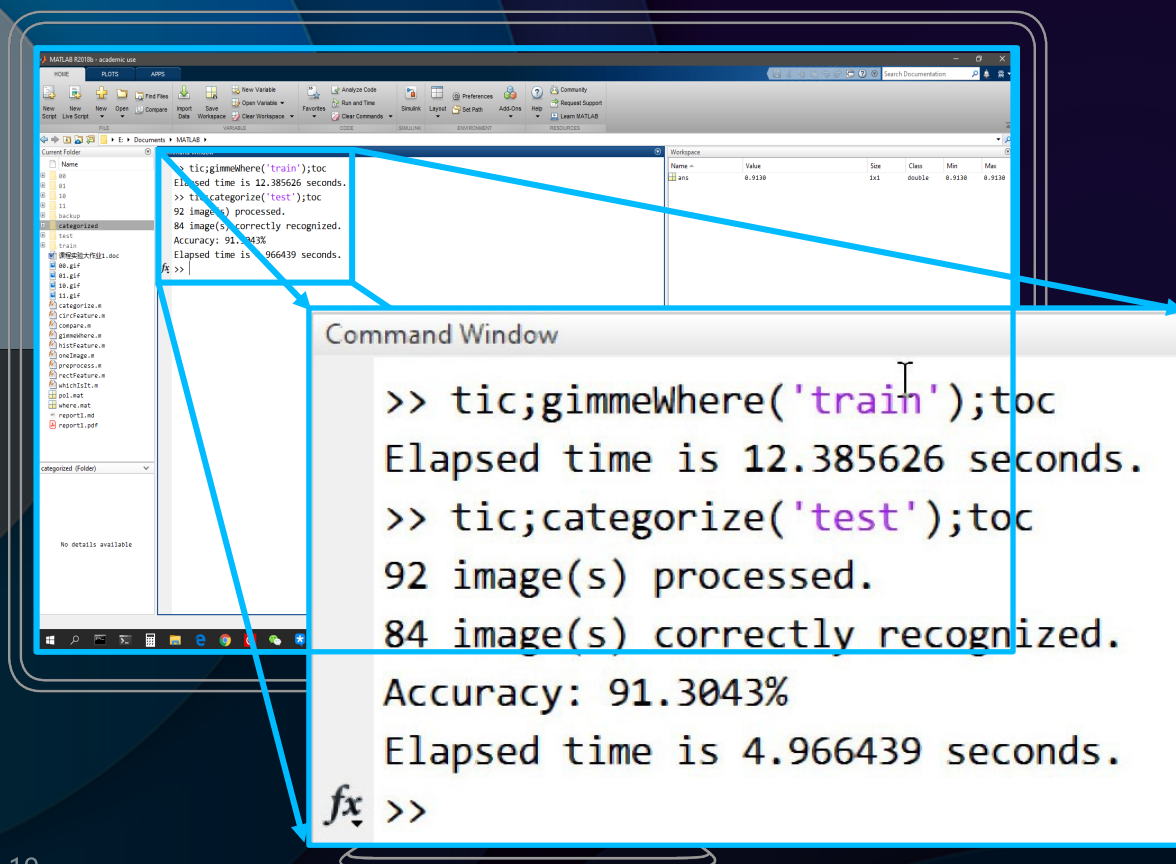
- 4. 单张图片特征比较
 - 4.1 向量相似度的比较方法
 - 4.2 总体相似度的获得 (三种特征相似度的整合)
 - 4.3 更高效的前k个最大值提取方法 ($O(n)$)
 - 4.4 复制以及重命名图片 (文件操作)
 - 5. 多张图片特征比较
 - 5.1 遍历 (文件夹操作、递归)
 - 5.2 总体类别的获得 (十个最相似图片信息整合)

目录

- 1. 预处理
 - 1.1 二值化
 - 1.2 缩放
 - 1.3 重心查找
 - 1.4 图片平移
- 2. 特征提取
 - 2.1 矩形拆分特征 (直角坐标)
 - 2.2 扇形拆分特征 (极坐标)
 - 2.3 直方图
- 3. 数据库变量构建
 - 3.1 遍历 (文件夹操作、递归)
 - 3.2 信息整合与储存
- 4. 单张图片特征比较
 - 4.1 向量相似度的比较方法
 - 4.2 总体相似度的获得 (三种特征相似度的整合)
 - 4.3 更高效的前k个最大值提取方法 ($O(n)$)
 - 4.4 复制以及重命名图片 (文件操作)
- 5. 多张图片特征比较
 - 5.1 遍历 (文件夹操作、递归)
 - 5.2 总体类别的获得 (十个最相似图片信息整合)

运行结果

很抱歉我太菜了我没有UI



The screenshot shows the MATLAB R2018b interface. A callout box highlights the Command Window output, which contains the following text:

```
>> tic;gimmeWhere('train');toc  
Elapsed time is 12.385626 seconds.  
>> tic;categorize('test');toc  
92 image(s) processed.  
84 image(s) correctly recognized.  
Accuracy: 91.3043%  
Elapsed time is 4.966439 seconds.  
fx >>
```



让我解释一下数据集和测试集都是啥

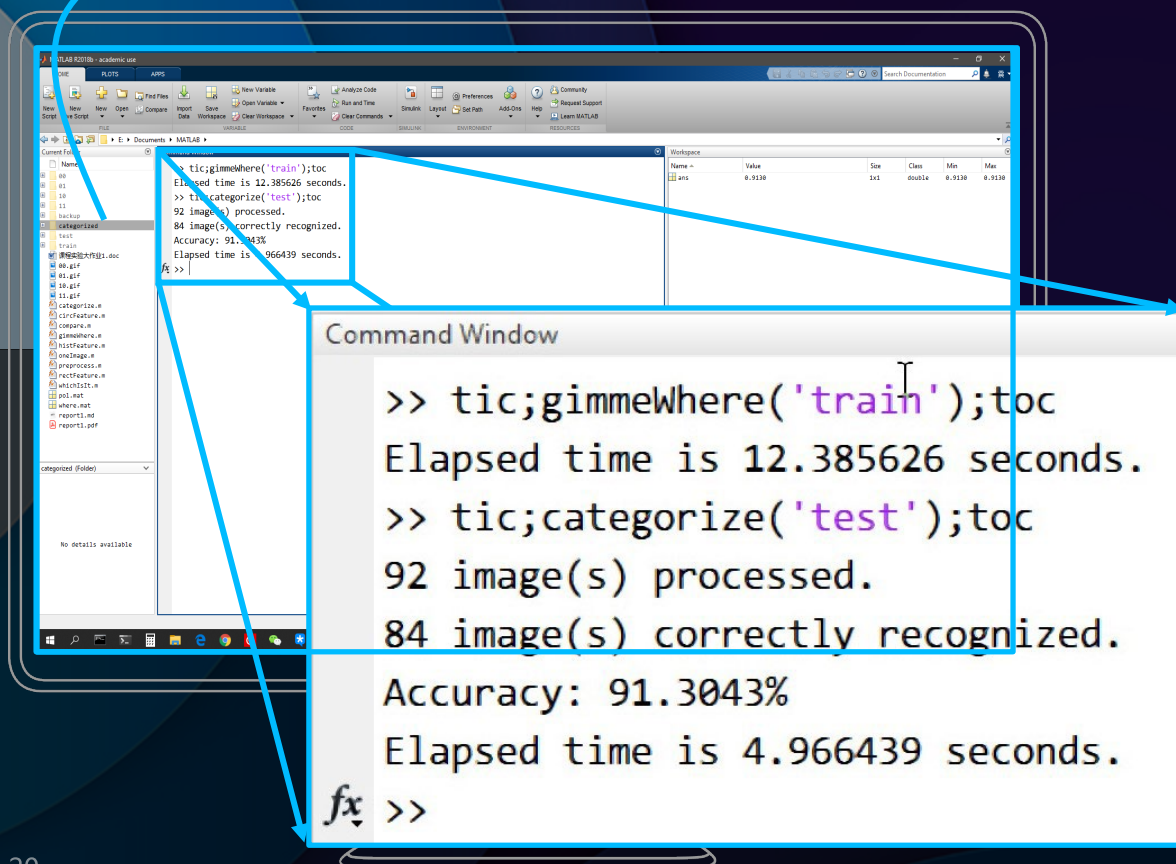
所谓的'train'是用于测试的数据集文件的文件夹，，是从完整数据集删除四分之一图片得来的。

我是用脚本删除，没有人工干预。

所谓的'test' 是用于测试的测试图片文件夹，保留了上述删除的图片，删除上述train中的图片，由此两个数据集中没有重复图片。

运行结果

很抱歉我太菜了我没有UI



The screenshot shows the MATLAB R2018b interface. A callout box highlights the Command Window output, which contains the following text:

```
>> tic;gimmeWhere('train');toc  
Elapsed time is 12.385626 seconds.  
>> tic;categorize('test');toc  
92 image(s) processed.  
84 image(s) correctly recognized.  
Accuracy: 91.3043%  
Elapsed time is 4.966439 seconds.  
fx >>
```



让我解释一下数据集和测试集都是啥

所谓的'train'是用于测试的数据集文件的文件夹，是从完整数据集删除四分之一图片得来的。

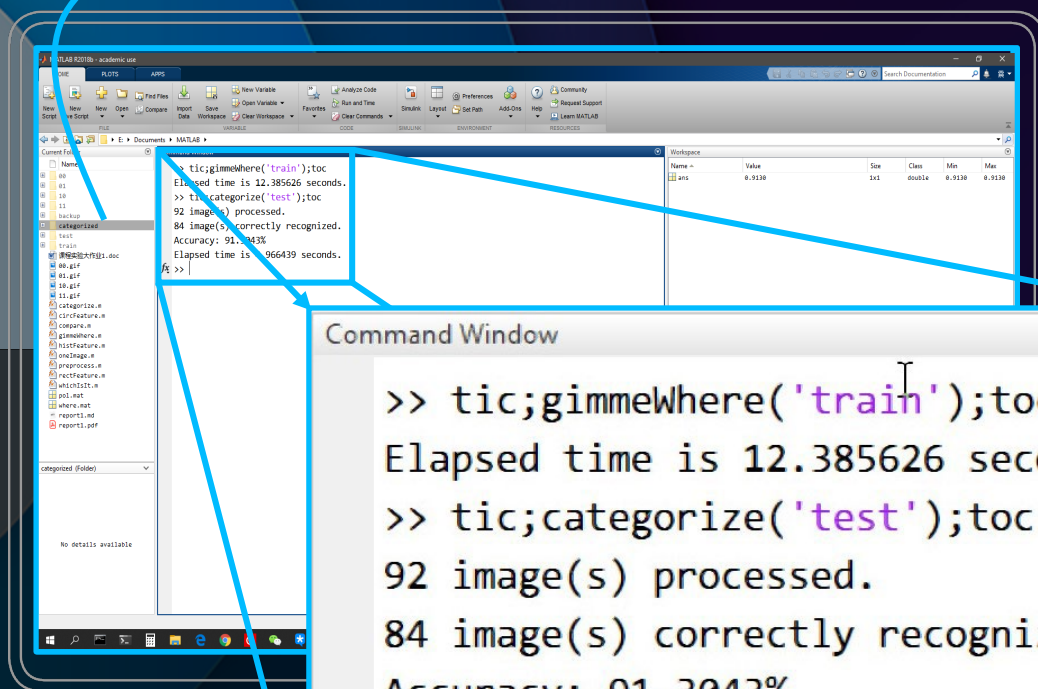
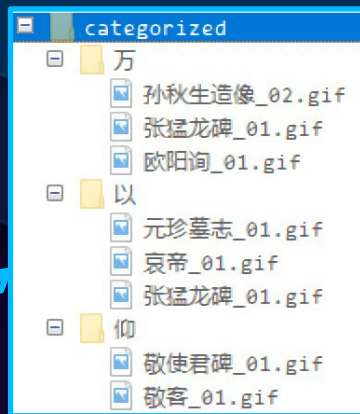
我是用脚本删除，没有人工干预。



所谓的'test' 是用于测试的测试图片文件夹，保留了上述删除的图片，删除上述train中的图片，由此两个数据集中没有重复图片。

运行结果

很抱歉我太菜了我没有UI



Command Window

```
>> tic;gimmeWhere('train');toc  
Elapsed time is 12.385626 seconds.  
>> tic;categorize('test');toc  
92 image(s) processed.  
84 image(s) correctly recognized.  
Accuracy: 91.3043%  
Elapsed time is 4.966439 seconds.
```

fx >>



让我解释一下数据集和测试集都是啥

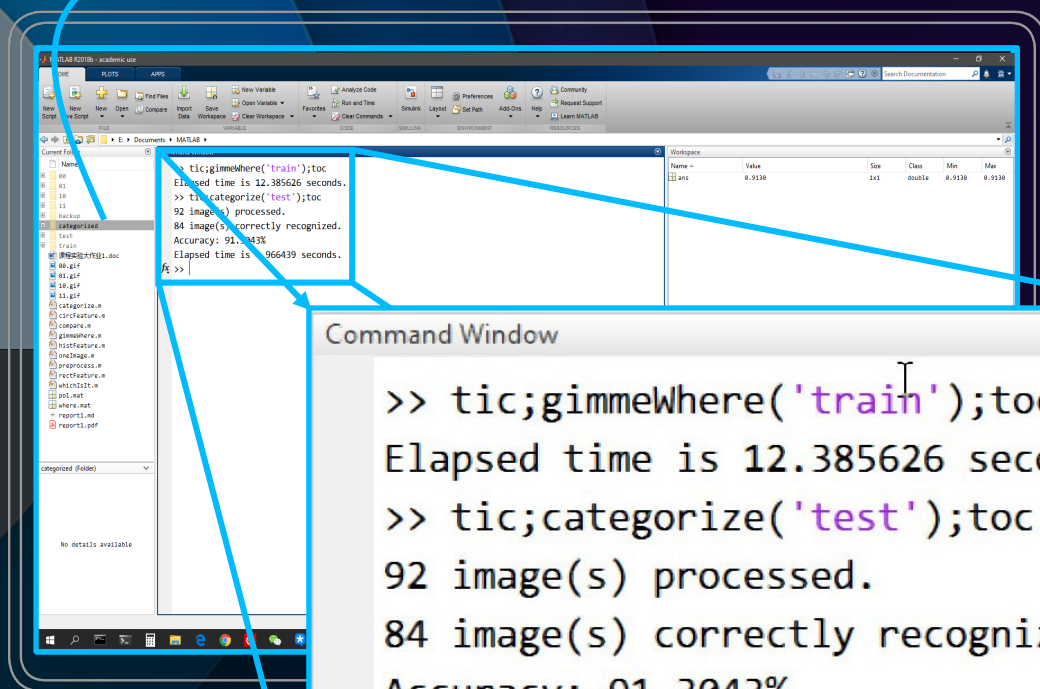
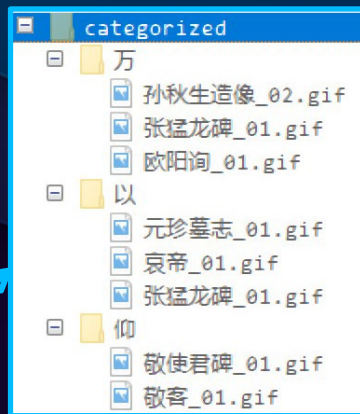
所谓的'train'是用于测试的数据集文件的文件夹，是从完整数据集删除四分之一图片得来的。

我是用脚本删除，没有人工干预。

所谓的'test' 是用于测试的测试图片文件夹，保留了上述删除的图片，删除上述train中的图片，由此两个数据集中没有重复图片。

运行结果

很抱歉我太菜了我没有UI



Command Window

```
>> tic;gimmeWhere('train');toc  
Elapsed time is 12.385626 seconds.  
>> tic;classify('test');toc  
92 image(s) processed.  
84 image(s) correctly recognized.  
Accuracy: 91.3043%  
Elapsed time is 4.966439 seconds.
```

fx >>

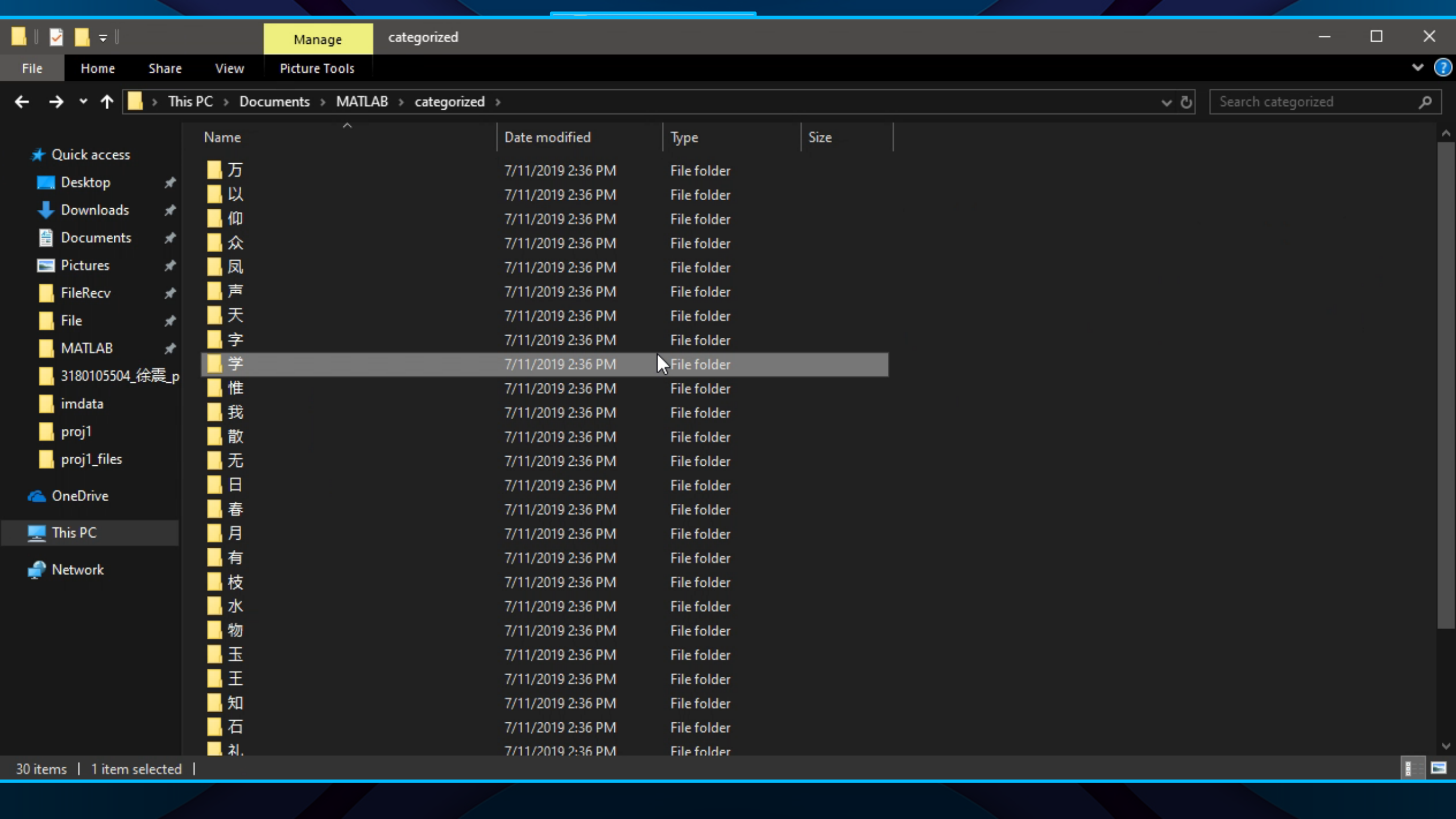


让我解释一下数据集和测试集都是啥

所谓的'train'是用于测试的数据集文件的文件夹，是从完整数据集删除四分之一图片得来的。

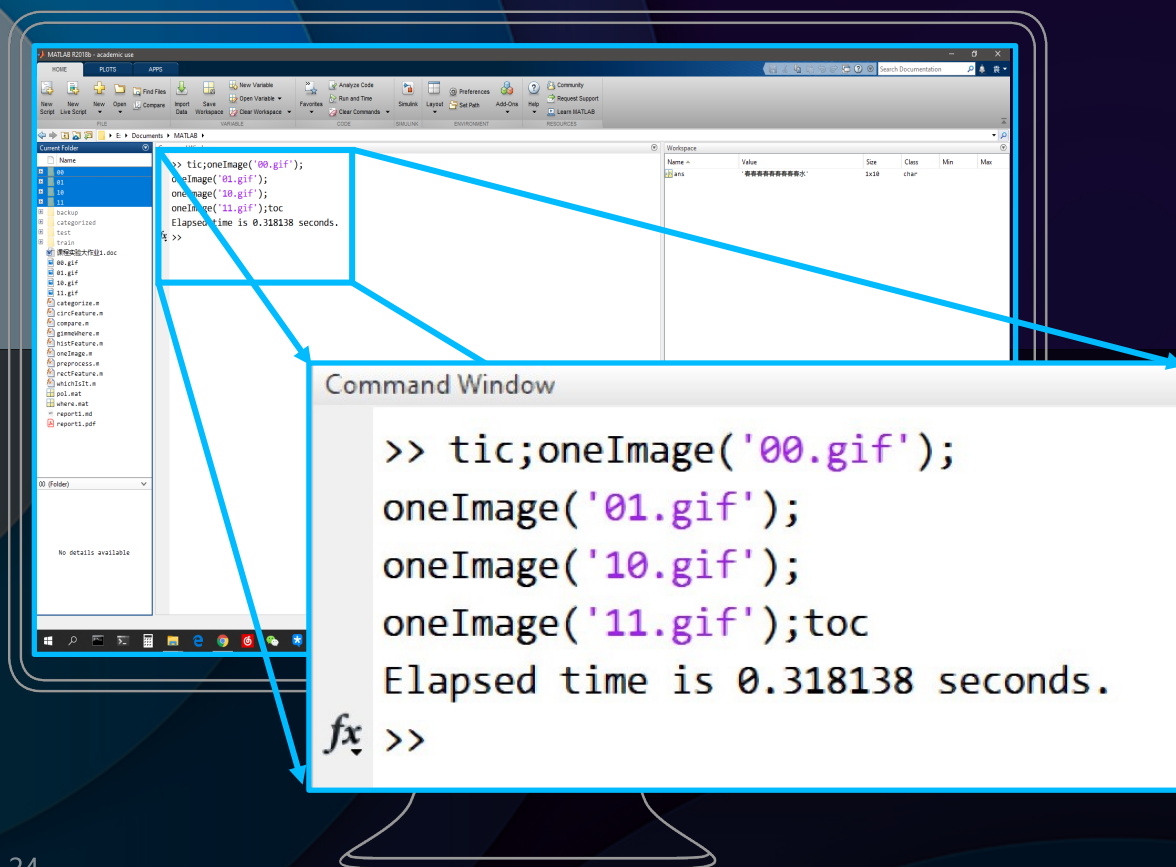
我是用脚本删除，没有人工干预。

所谓的'test' 是用于测试的测试图片文件夹，保留了上述删除的图片，删除上述train中的图片，由此两个数据集中没有重复图片。



运行结果

很抱歉我太菜了我没有UI



单张图片的处理

- 本函数接收工作目录下的某张图片的文件名（包含拓展名）。请直接输入文件名例如 '00.gif'，不要输入 '\\00.gif'。
- 本函数没有返回值，在工作目录下创建以文件名称（无拓展名）为名的目录，将该图片复制进该目录，并从数据库（'backup'）中复制相似度前十的图片到该目录下，数据库中的图片名称以相似度开头。

FileHomeShareViewPicture Tools

←→↕↑

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- FileRecv
- File
- MATLAB
- 3180105504_徐震_p
- imdata
- proj1
- proj1_files

OneDriveThis PCNetwork

ManageMATLAB

This PC > Documents > MATLAB

Search MATLAB

00011011backupcategorizedtesttrain00.gif01.gif10.gif11.gif

categorize.mcircFeature.mcompare.mgimmeWhere.mhistFeature.monelImage.mpol.matpreprocess.mrectFeature.mreport1.mdreport1.pdfwhere.mat

whichIsIt.m课程实验大作业1.doc

26 items | 4 items selected 9.28 KB

00

categorize.m

01

circFeature.m

10

compare.m

11

gimmeWhere.m

backup

histFeature.m

categorized

onelImage.m

test

pol.mat

train

preprocess.m

00.gif

rectFeature.m

01.gif

report1.md

10.gif

report1.pdf

11.gif

where.mat

whichIsIt.m

课程实验大
作业1.doc

预处理

- 二值化
- 缩放
- 重心查找和平移

预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



预处理

- 二值化
- 缩放
- 重心查找和平移



特征提取

- 直角坐标系下的特征提取
- 极坐标系下的特征提取
- 直方图与汉字识别

特征提取

- 直角坐标系下的特征提取
- 极坐标系下的特征提取
- 直方图与汉字识别



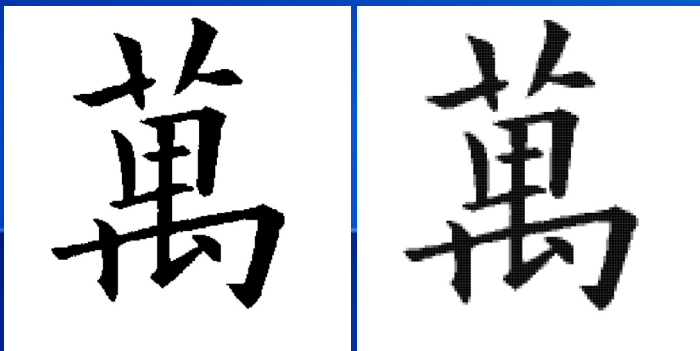
特征提取

- 直角坐标系下的特征提取
- 极坐标系下的特征提取
- 直方图与汉字识别



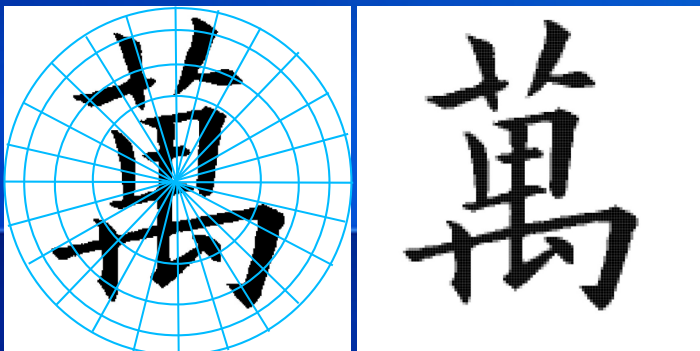
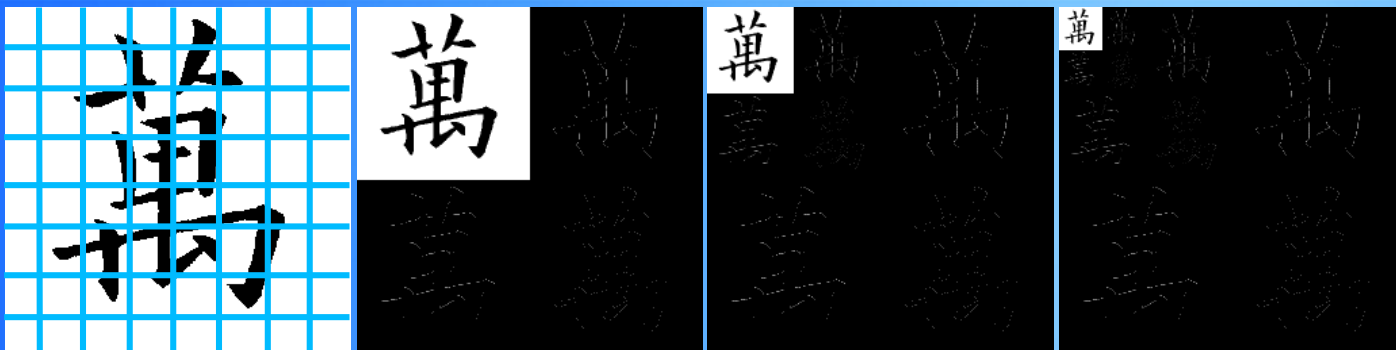
特征提取

- 直角坐标系下的特征提取
- 极坐标系下的特征提取
- 直方图与汉字识别



特征提取

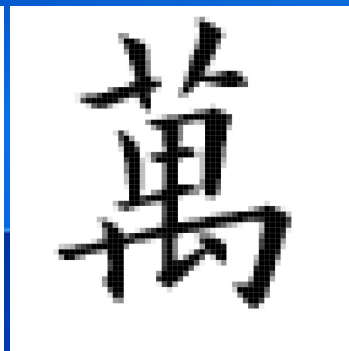
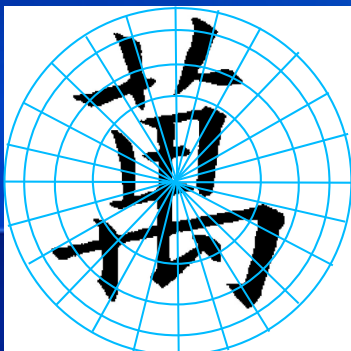
- 直角坐标系下的特征提取
- 极坐标系下的特征提取
- 直方图与汉字识别



```

1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat -- 存储特征信息的矩阵
7     ... feat = zeros(num1, num2);
8
9     ... for i = 1:num1
10        ... for j = 1:num2
11            ... feat(i, j) = sum(sum(bw(i * 256 / num1 - (256 / num1 - 1):i * 256 / num1, j * 256 / num2 - (256 / num2 - 1):j * 256 / num2)));
12        ... end
13    ... end
14    ... % 归一化操作, 实际上不需要
15    ... % feat = feat ./ (256 * 256 ./ (num1 * num2));
16

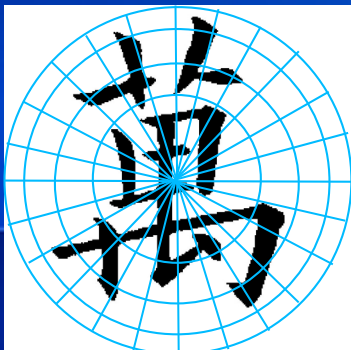
```



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```



45

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19
20                 ... for l = 1:step
21
22                     ... if (bw(i, j))
23
24                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
25                             ... feat(k, l) = feat(k, l) + 1;
26                         ... end
27                     ... end
28                 ... end
29             ... end
30         ... end
31     ... end
32
33     ... end
34
35     ... end
36
```

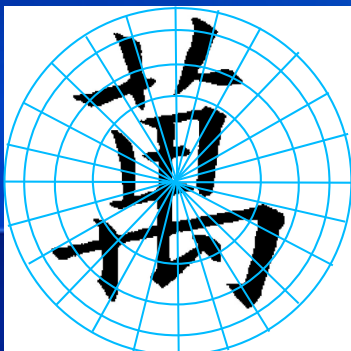
6 / num2 - 1): j * 256 / num2));



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```



46

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

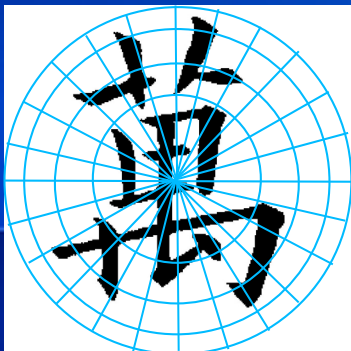
6 ./ num2 - 1): j * 256 ./ num2));



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```



47

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

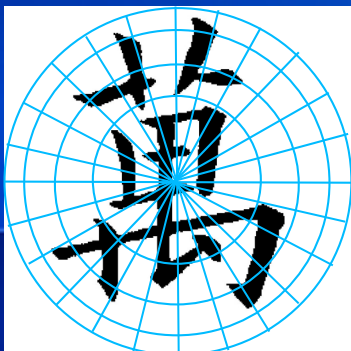
6 / num2 - 1): j * 256 / num2));



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```



48

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

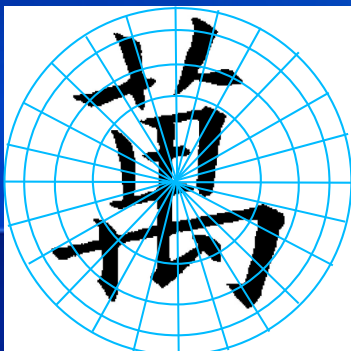
6 / num2 - 1): j * 256 / num2));



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```



49

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

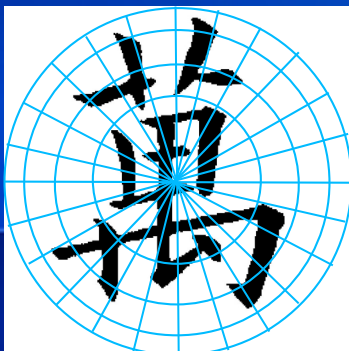
6 / num2 - 1): j * 256 / num2));



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

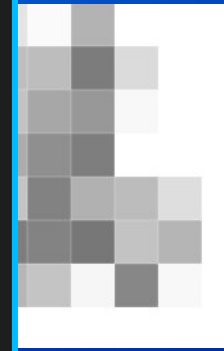


50

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

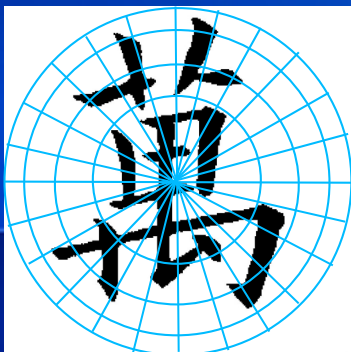
6 / num2 - 1): j * 256 / num2));



```

1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16

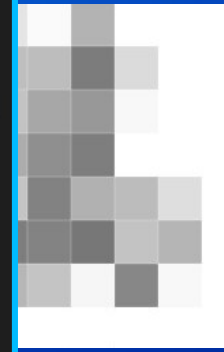
```



```

1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end

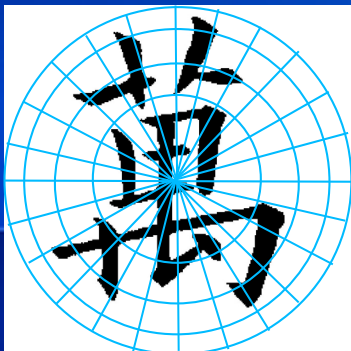
```

$$6 ./ \text{num2} - 1): j * 256 ./ \text{num2}));$$


rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat -- 存
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

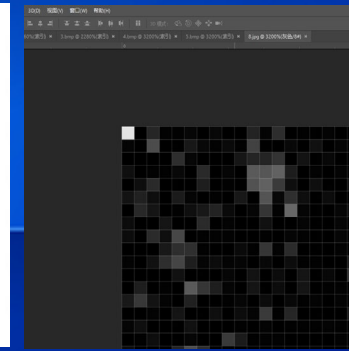
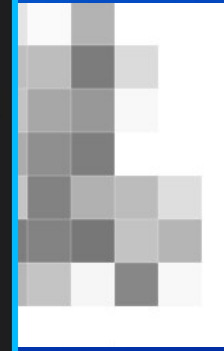


52

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ...
20                 ... for l = 1:step
21                     ...
22                     ... if (bw(i, j))
23                         ...
24                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
25                             ... feat(k, l) = feat(k, l) + 1;
26                         ... end
27                     ... end
28                 ... end
29             ... end
30         ... end
31     ... end
32
33     ... end
34
35     ... end
36
```

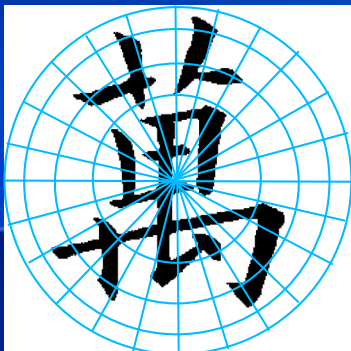
6 / num2 - 1): j * 256 / num2));



rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

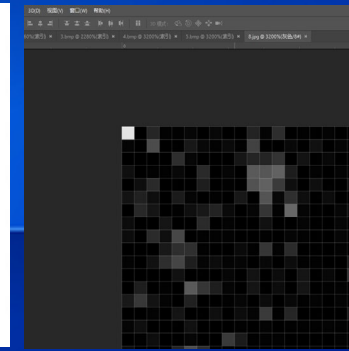
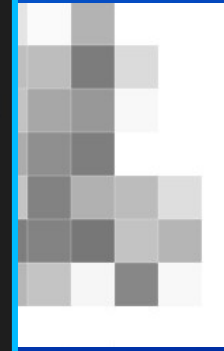


53

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ...
20                 ... for l = 1:step
21                     ...
22                     ... if (bw(i, j))
23                         ...
24                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
25                             ... feat(k, l) = feat(k, l) + 1;
26                         ... end
27                     ... end
28                 ... end
29             ... end
30         ... end
31     ... end
32
33     ... end
34
35     ... end
36
```

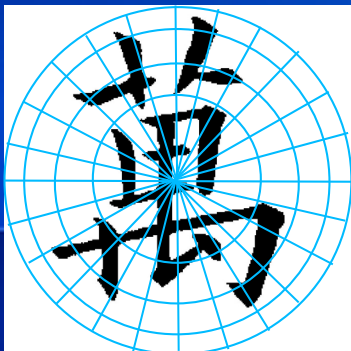
6 / num2 - 1): j * 256 / num2));




```

1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16

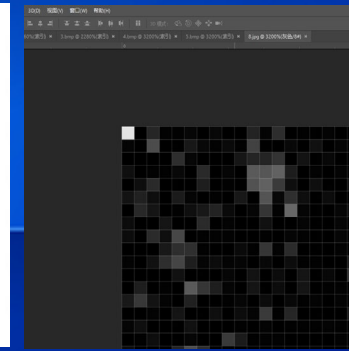
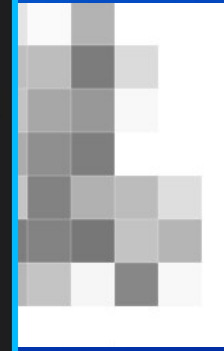
```



```

1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end

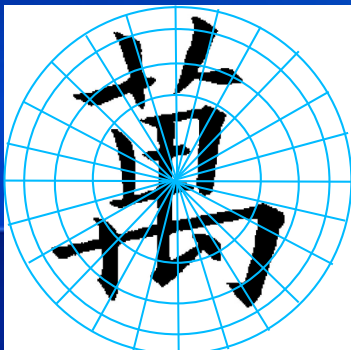
```

$$6 ./ \text{num2} - 1): j * 256 ./ \text{num2})))$$


rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat -- 存
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

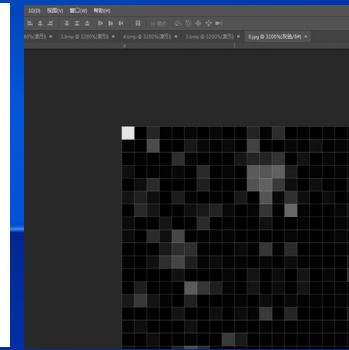
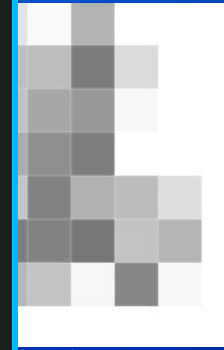


55

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

6 ./ num2 - 1): j * 256 ./ num2)))



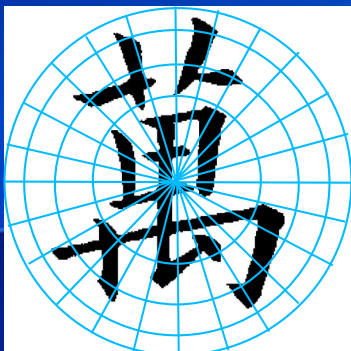
车

直方图

rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat 存在
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

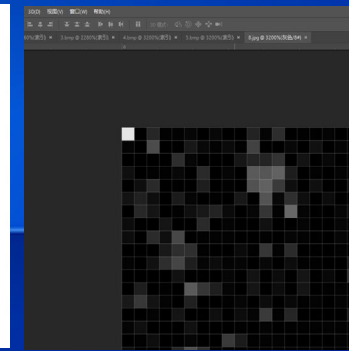
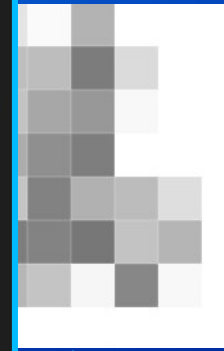


56

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20
21                     ... if (bw(i, j))
22                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
23                             ... feat(k, l) = feat(k, l) + 1;
24                         ... end
25                     ... end
26                 ... end
27             ... end
28         ... end
29     ... end
30
31     ... end
32
33     ... end
34
35     ... end
36
```

6 / num2 - 1): j * 256 / num2));



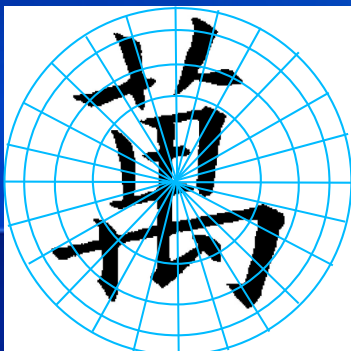
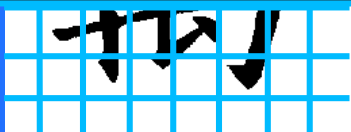
车

直方图

rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat -- 存
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

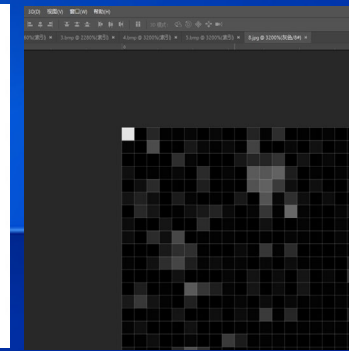
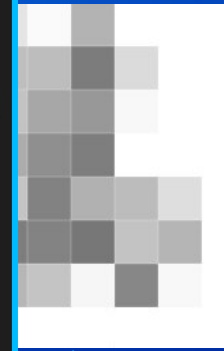


57

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

6 ./ num2 - 1): j * 256 ./ num2)))



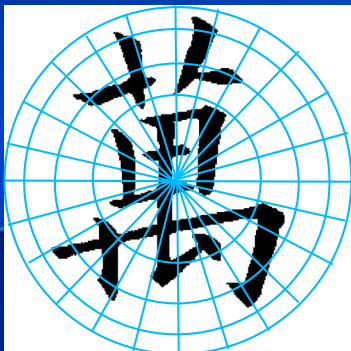
车

直方图

rectFeature.m ✕

rectFeature.m ▸ ...

```
1 function feat = rectFeature(bw)
2     ... % 矩形 (直角坐标系) 的特征提取函数, 请传入BW图像矩阵 (logical)
3     ... % 两个数字8是开发者测试得到的结果
4     ... num1 = 8;
5     ... num2 = 8;
6     ... % feat -- 存
7     ... feat = zeros(4, num1 + num2);
8
9     ... for i = 1:
10         ... for j =
11             ... fe
12         ... end
13     ... end
14     ... % 归一化操作
15     ... % feat = f
16
```

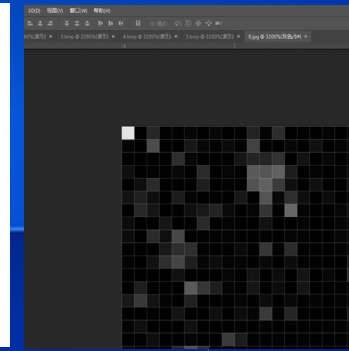
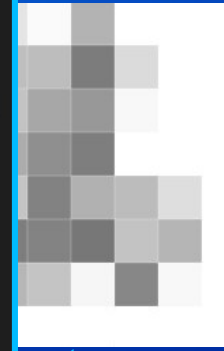


58

circFeature.m ▸ ...

```
1 function feat = circFeature(bw)
2     ... % 扇形特征 (极坐标系) 提取函数, 请传入BW图像矩阵
3     ... length = 256;
4     ... step = 20;
5     ... feat = zeros(4, step);
6     ... rarr = [1, 74, 91, 111, 128];
7     ... thetarr = [0:2 * pi / step:2 * pi, 2 * pi + eps];
8
9     ... for i = 1:length
10
11         ... for j = 1:length
12             ... [theta, r] = cart2pol(i - length / 2, j - length / 2);
13
14             ... if theta < 0
15                 ... theta = theta + 2 * pi;
16             ... end
17
18             ... for k = 1:4
19                 ... for l = 1:step
20                     ... if (bw(i, j))
21                         ... if rarr(k) ≤ r && r < rarr(k + 1) && thetarr(l) ≤ theta && theta < thetarr(l + 1)
22                             ... feat(k, l) = feat(k, l) + 1;
23                         ... end
24                     ... end
25                 ... end
26             ... end
27         ... end
28     ... end
29
30     ... end
31
32     ... end
33
34     ... end
35
36     ... end
```

6 ./ num2 - 1): j * 256 ./ num2)))



车

直方图

工作簿1 - Excel

文件 开始 插入 页面布局 公式 数据 审阅 视图 Acrobat Team 告诉我想要做什么...

剪贴板

粘贴 格式刷

字体

对齐方式

数字

样式

单元格

编辑

1246				✕ ✓ f _x																																			
	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC				
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.75	0.75	1	1	1	1	1	1	1	1	1	1	1	1	1					
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.75	0	0	0.5	1	1	1	1	1	1	1	1	1	1	1					
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.25	0	0	0	0	0.5	0.75	1	1	1	1	1	1	1					
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.75	0	0	0	0	0	0	0	0.25	1	1	1	1	1					
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.5	0	0	0	0	0	0	0	0	1	1	1	1					
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.75	0	0	0	0	0	0	0	0	0.5	1	1	1					
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0.5	1	1	1					
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.5	0	0	0	0	0	0	0	1	1	1					
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0.5	1	1	1					
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1					
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.5	0	0	0	0	0	0	1	1	1					

就绪

Sheet1

85%

通过特征矩阵计算相似度

- 巴氏距离
- 改良的巴氏距离

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$

A square box containing a handwritten Chinese character, '學' (Xue), written in black ink on a white background. The character is written in a cursive, calligraphic style.

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多

特征矩阵1没那么多

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多



特征矩阵1没那么多



特征矩阵里1没那么多

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多



特征矩阵1没那么多



特征矩阵里1没那么多

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

5.4020e+04

	1	2	3	4	5	6	7	8
1	1024	1024	1010	1024	694	1024	1024	1024
2	1024	1024	554	803	130	786	1024	1024
3	1024	1009	511	466	334	810	1024	1024
4	1024	1024	440	75	219	769	1024	1024
5	1024	926	629	85	264	499	642	1024
6	997	428	166	131	623	759	360	1024
7	1024	1016	788	891	1024	505	927	1024
8	1024	1024	1024	1024	1024	1024	1024	1024

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

	1	2	3	4	5	6	7	8		
1	1024	1024	994	1024	835	830	1024	1024		
2	1024	1019	610	723	572	256	1014	1024		
3	1024	813	336	424	188	774	1021	1024	7	8
4	1024	1024	424	131	126	638	1024	1024	1024	1024
5	1024	1024	561	130	269	574	988	1024	1024	1024
6	1016	423	163	107	181	374	630	1024	1024	1024
7	1024	1018	873	831	740	279	951	1024	1024	1024
8	1024	1024	1024	1024	1024	1024	1024	1024	642	1024
		6	997	428	166	131	623	759	360	1024
		7	1024	1016	788	891	1024	505	927	1024
		8	1024	1024	1024	1024	1024	1024	1024	1024

5.4020e+04

4.9863e+04

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

7	8
1024	1024
1024	1024
1024	1024
642	1024

5	6	7	8
1024	1024	561	130
1016	423	163	107
1024	1018	873	831
1024	1024	1024	1024

6	7	8
997	428	166
1024	1016	788
1024	1024	1024

6	7	8
130	269	574
630	1024	1024
951	1024	1024
1024	1024	1024

6	7	8
623	759	360
1024	505	927
1024	1024	1024

特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

5.4020e+04

4.9863e+04

左边两张图比右边两张更像?

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

7	8
1024	1024
1024	1024
1024	1024
642	1024

5	6	7	8
1024	1024	561	130
1016	423	163	107
1024	1018	873	831
1024	1024	1024	1024

6	7	8
997	428	166
1024	1016	788
1024	1024	1024

6	7	8
130	269	574
630	1024	1024
951	1024	1024
1024	1024	1024

6	7	8
623	759	360
1024	505	927
1024	1024	1024

特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

5.4020e+04

4.9863e+04

左边两张图比右边两张更像?

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

7	8
1024	1024
1024	1024
1024	1024
642	1024

5	6	7	8
1024	1024	561	130
1016	423	163	107
1024	1018	873	831
1024	1024	1024	1024

6	7	8
997	428	166
1024	1016	788
1024	1024	1024

6	7	8
130	269	574
630	1024	1024
951	1024	1024
1024	1024	1024

6	7	8
623	759	360
1024	505	927
1024	1024	1024

特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

5.4020e+04

4.9863e+04

左边两张图比右边两张更像?

通过特征矩阵计算相似度

- 巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \sqrt{p_i \cdot q_i}$$



	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024

7	8
1024	1024
1024	1024
1024	1024
1024	1024

5	6	7	8
1024	1024	561	130
1024	1024	269	574
1024	1024	988	1024
1024	1024	1024	1024
1024	1024	1024	1024
1024	1024	1024	1024
1024	1024	1024	1024

6	7	8
997	428	166
131	623	759
360	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024

特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

5.4020e+04

4.9863e+04

左边两张图比右边两张更像?

通过特征矩阵计算相似度

- 改良的巴氏距离

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多

特征矩阵1没那么多

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多



特征矩阵1没那么多



特征矩阵里1没那么多

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多



特征矩阵1没那么多



特征矩阵里1没那么多

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多



特征矩阵1没那么多

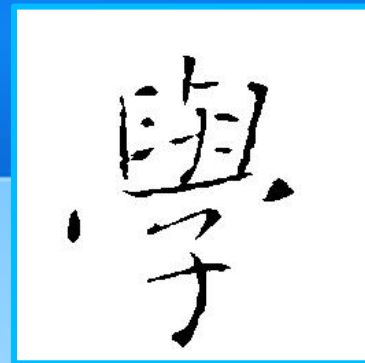


特征矩阵里1没那么多

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

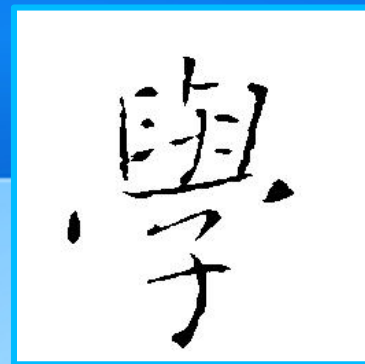
0.483593

	1	2	3	4	5	6	7	8
1	1024	1024	1010	1024	694	1024	1024	1024
2	1024	1024	554	803	130	786	1024	1024
3	1024	1009	511	466	334	810	1024	1024
4	1024	1024	440	75	219	769	1024	1024
5	1024	926	629	85	264	499	642	1024
6	997	428	166	131	623	759	360	1024
7	1024	1016	788	891	1024	505	927	1024
8	1024	1024	1024	1024	1024	1024	1024	1024

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

	1	2	3	4	5	6	7	8
1	1024	1024	994	1024	835	830	1024	1024
2	1024	1019	610	723	572	256	1014	1024
3	1024	813	336	424	188	774	1021	1024
4	1024	1024	424	131	126	638	1024	1024
5	1024	1024	561	130	269	574	988	1024
6	1016	423	163	107	181	374	630	1024
7	1024	1018	873	831	740	279	951	1024
8	1024	1024	1024	1024	1024	1024	1024	1024

7	8
1024	1024
1024	1024
1024	1024
1024	1024
642	1024

6	997	428	166	131	623	759	360	1024
7	1024	1016	788	891	1024	505	927	1024
8	1024	1024	1024	1024	1024	1024	1024	1024

0.483593

0.496283

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$

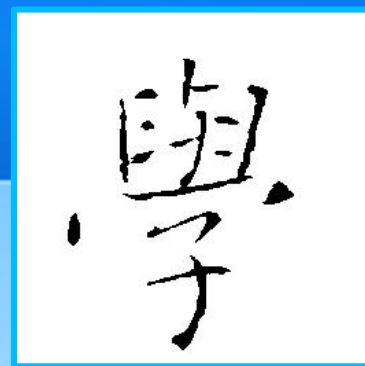
	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

7	8
1024	1024
1016	423
1024	1018
1024	1024
997	428
1024	1016
1024	1024
1024	1024

6	7	8
130	269	574
163	181	374
873	831	740
166	131	623
788	891	1024
1024	1024	1024
1024	1024	1024
642	1024	1024

6	7	8
561	130	269
163	181	374
873	831	740
166	131	623
788	891	1024
1024	1024	1024
1024	1024	1024
642	1024	1024



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

0.483593

0.496283

右边两张图比左边两张更像!

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$

	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

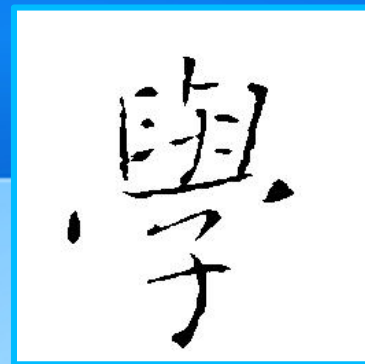
7	8
1024	1024
1016	423
1024	1018
1024	1024
997	428
1024	1016
1024	1024
1024	1024

6	7	8
1024	1024	1024
1016	423	163
1024	1018	873
1024	1024	1024
997	428	166
1024	1016	788
1024	1024	1024
1024	1024	1024

6	7	8
1024	1024	1024
1016	423	163
1024	1018	873
1024	1024	1024
997	428	166
1024	1016	788
1024	1024	1024
1024	1024	1024

6	7	8
1024	1024	1024
1016	423	163
1024	1018	873
1024	1024	1024
997	428	166
1024	1016	788
1024	1024	1024
1024	1024	1024

6	7	8
1024	1024	1024
1016	423	163
1024	1018	873
1024	1024	1024
997	428	166
1024	1016	788
1024	1024	1024
1024	1024	1024



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

0.483593

0.496283

右边两张图比左边两张更像!

通过特征矩阵计算相似度

- 改良的巴氏距离

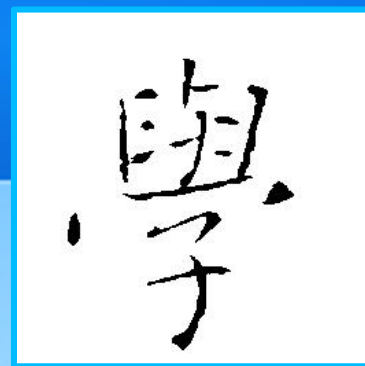
$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$

	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

7	8
1024	1024
1016	423
1024	1018
1024	1024
997	428
1024	1016
1024	1024
1024	1024

6	7	8
1024	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024
1024	1024	1024



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

0.483593

0.496283

右边两张图比左边两张更像!

通过特征矩阵计算相似度

- 改良的巴氏距离

$$\rho(p, q) = \sum_{i=1}^n \frac{\sqrt{p_i \cdot q_i}}{\sum_i^n p_i + \sum_i^n q_i}$$

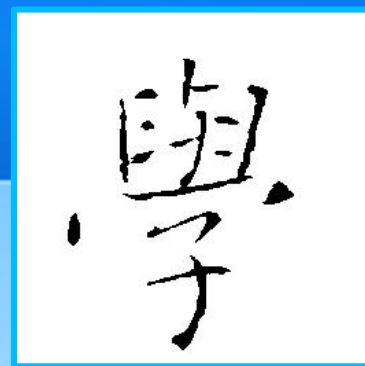
	1	2	3	4	5	6	7	8
1	1024	1024	1024	1024	1024	1024	1024	1024
2	1024	1024	971	825	938	914	1024	1024
3	1024	1024	878	958	836	837	1024	1024
4	1024	1024	810	917	811	824	1009	1024
5	1024	935	960	819	844	916	952	1024
6	1024	951	1020	963	821	1024	1024	1024
7	1024	1024	1022	956	823	1024	1024	1024
8	1024	1024	1024	863	986	1024	1024	1024

7	8
1024	1024
1014	1024
1021	1024
1024	1024
988	1024
1024	1024
1024	1024
1024	1024

7	8
1024	1024
1016	423
1024	1018
1024	1024
997	428
1024	1016
1024	1024
1024	1024

6	7	8
130	269	574
107	181	374
873	831	740
166	131	623
788	891	1024
1024	1024	1024
1024	1024	1024
642	1024	1024

6	7	8
561	130	269
163	107	181
873	831	740
166	131	623
788	891	1024
1024	1024	1024
1024	1024	1024
360	1024	1024



特征矩阵1很多

特征矩阵1没那么多

特征矩阵里1没那么多

0.483593

0.496283

右边两张图比左边两张更像!

谢谢大家