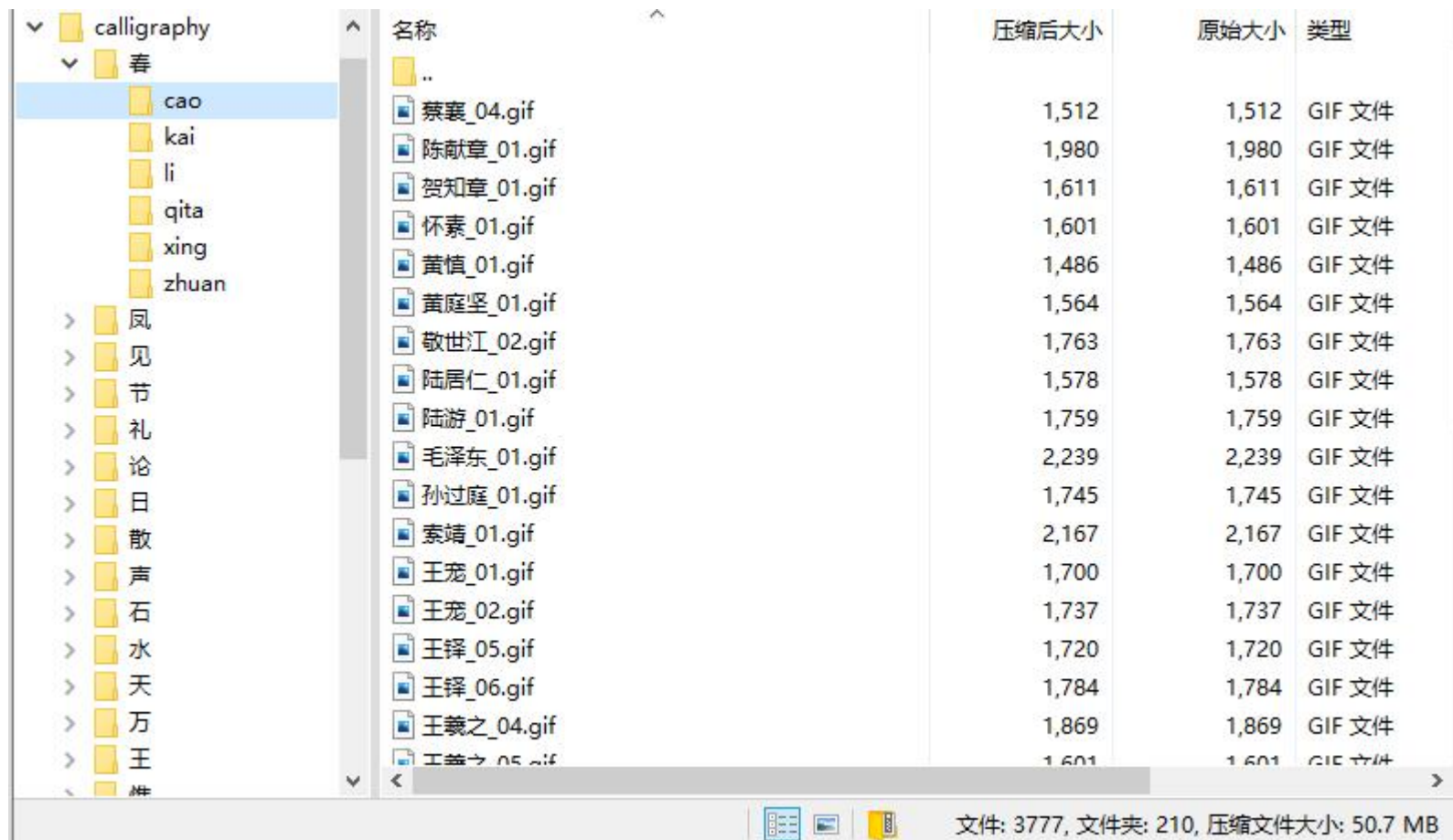


课程实验大作业之一：

- 相似书法字检索：



Project1 素材已上传到FTP:
课件/calligraphy.rar



The screenshot shows a file explorer window. On the left, a tree view shows the directory structure: 'calligraphy' > '春' > 'cao' (selected). Below 'cao' are subdirectories 'kai', 'li', 'qita', 'xing', and 'zhuan'. To the right of the tree, a list of files is displayed with columns for '名称' (Name), '压缩后大小' (Compressed Size), '原始大小' (Original Size), and '类型' (Type). The files are all GIF files, mostly named after Chinese calligraphers. The status bar at the bottom indicates '文件: 3777, 文件夹: 210, 压缩文件大小: 50.7 MB'.

名称	压缩后大小	原始大小	类型
..			
蔡襄_04.gif	1,512	1,512	GIF 文件
陈献章_01.gif	1,980	1,980	GIF 文件
贺知章_01.gif	1,611	1,611	GIF 文件
怀素_01.gif	1,601	1,601	GIF 文件
黄慎_01.gif	1,486	1,486	GIF 文件
黄庭坚_01.gif	1,564	1,564	GIF 文件
敬世江_02.gif	1,763	1,763	GIF 文件
陆居仁_01.gif	1,578	1,578	GIF 文件
陆游_01.gif	1,759	1,759	GIF 文件
毛泽东_01.gif	2,239	2,239	GIF 文件
孙过庭_01.gif	1,745	1,745	GIF 文件
索靖_01.gif	2,167	2,167	GIF 文件
王宠_01.gif	1,700	1,700	GIF 文件
王宠_02.gif	1,737	1,737	GIF 文件
王铎_05.gif	1,720	1,720	GIF 文件
王铎_06.gif	1,784	1,784	GIF 文件
王羲之_04.gif	1,869	1,869	GIF 文件
王羲之_05.gif	1,601	1,601	GIF 文件

文件: 3777, 文件夹: 210, 压缩文件大小: 50.7 MB

输入图片

散



$$\text{相似度} = \frac{\text{重合像素点数}}{\text{输入图片像素点数}}$$

图片库

散
聲
鳳
聲
...

计算相似度

0.95

0.84

0.32

0.92

...

取 top-K

返回结果

散
散
聲
...

1. 一个简单的做法：

输入图片

散

提取特征



x

$$s = f(x, x_i)$$

2. 更好一点的方法:

图片库

散
聲
鳳
散
...

提取特征



x1

x2

x3

x4

...

计算相似度



0.92

0.18

0.09

0.95

...

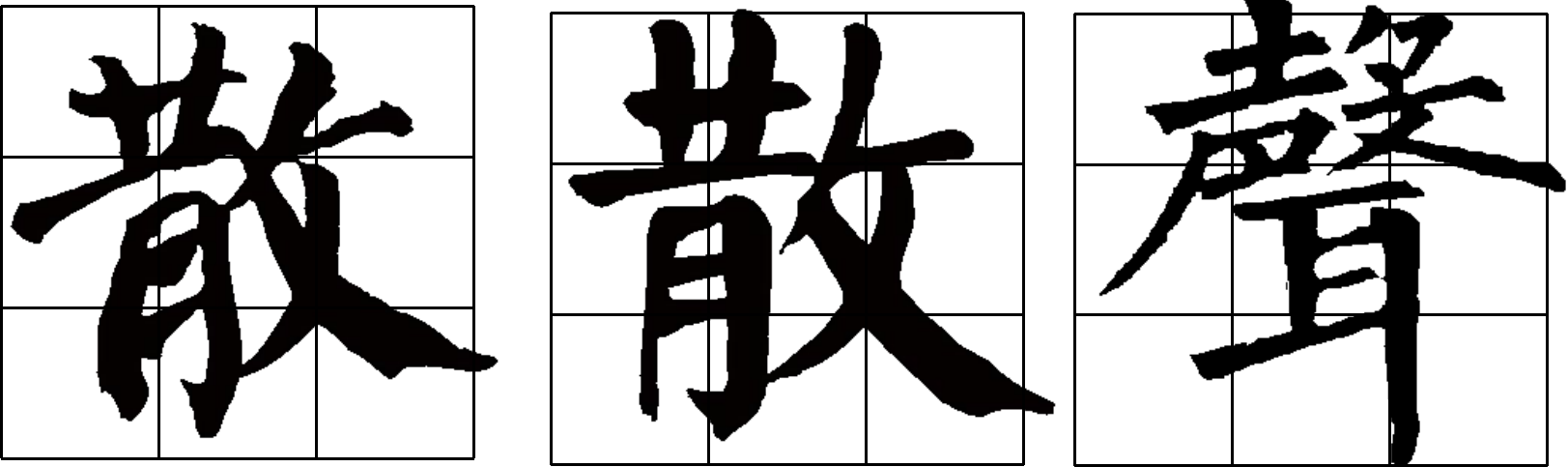
取 top-K



返回结果

散
散
散

提取特征：



特征向量

a b c d
x1 = [205, 9, 7, ...];
x2 = [215, 7, 8, ...];

例如：
特征a 代表轮廓像素点数；
特征b代表这个字有几横；
特征c代表这个字有几竖；
特征d

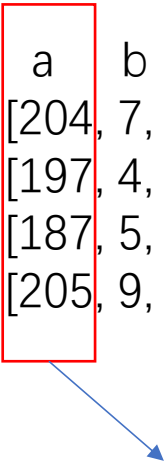
- 例如：
- 1. 轮廓的像素点数（边缘检测算法）
 - 2. 横竖笔画数（如何检测有几横几竖？）
 - 3. 3x3格点划分，每个格点的笔画分布？
 - 4. ...

余弦相似度

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}},$$

向量归一化、标准化:

	a	b	c	
x1 =	[204,	7,	7];	0~1 normalization
x2 =	[197,	4,	8];	
x3 =	[187,	5,	6];	
x4 =	[205,	9,	7];	



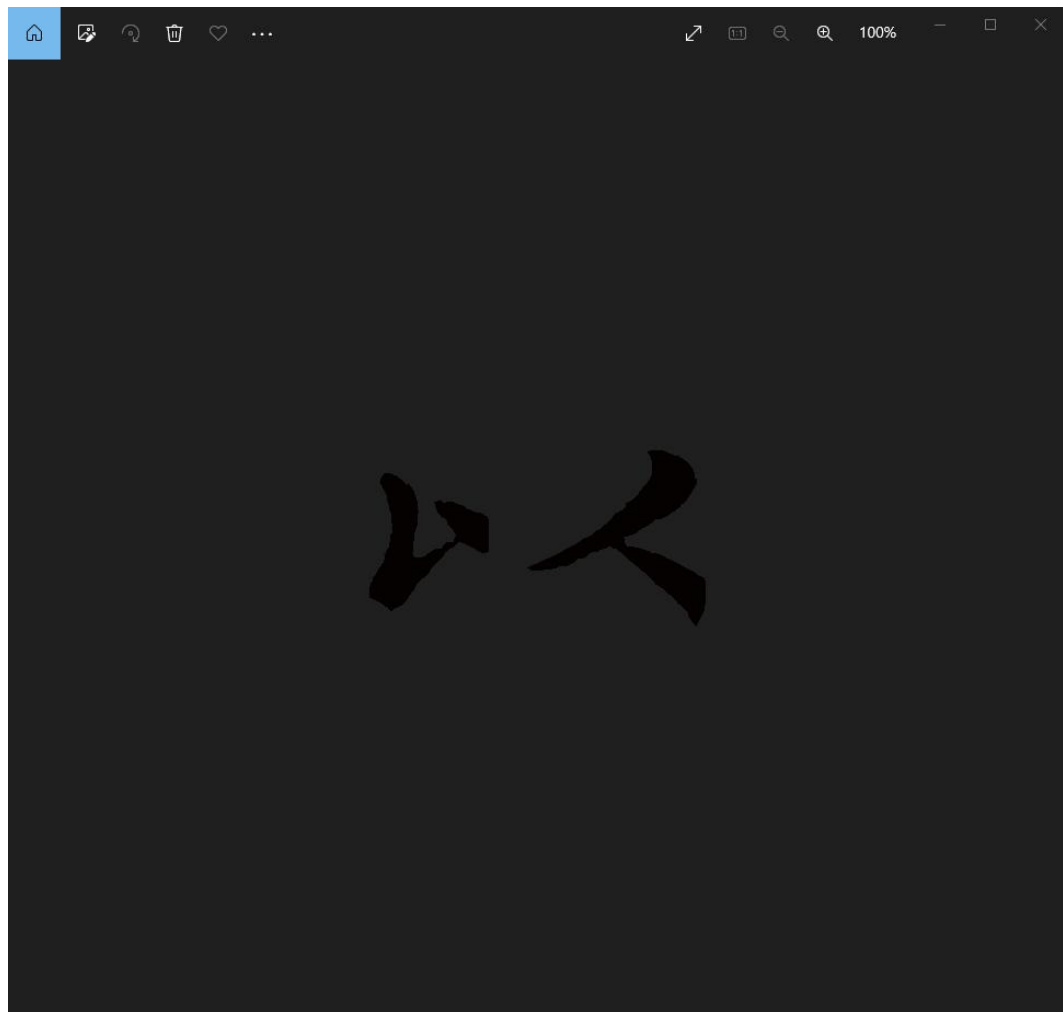
x1 = [0.9444, 0.6, 0.5]

0~1 normalization:
(归一化) $a = (a - \min_a) / (\max_a - \min_a)$

Standardization:
(标准化) $a = (a - \text{mean}_a) / \sigma_a$ σ_a : 标准差

一些问题

1. 显示很黑:



没关系，不影响识别。可以做直方图均衡，或者直接二值化。

大作业要求：

1. 对准确率的具体值没有要求，但是要有结果统计，例如：

4.2 测试样例

在与程序同个目录的文件夹下存有 8 个可供直接测试的图像，可以在第二个输入参数中直接输入这些图像的文件名；也可以输入 calligraphy 文件夹下的任意图像，但需输入完整路径和文件名。以下为其中 8 个测试样例。

输入图像	输入图像	测试结果	准确率
以/kai/袁帝_01.gif			100%
日/xing/黄庭坚_03.gif			65%
仰/kai/柳公权			70%

准确率统计：

例如，让程序返回相似度最高的前K张图片，K可以自定义。

例如K=20，这20张中正确的有16张，准确率就是80%

2. 不要将数据集打包进作业里提交，要让用户输入图片路径。（在命令行里输入图片路径），例如：

```
命令行窗口
>> calligraphy
请输入存储图片的文件夹路径：'D:\project\project1\calligraphy';
请输入待检索图像 路径：'D:\calligraphy\春\cao\黄庭坚_01.gif'
请输入希望检索出的图像个数：20
fx
```

使用genpath函数获取文件夹下所有子文件路径
通过寻找分隔符';'来寻找路径
读取所有子文件夹中的图片