

# 71086032-曾诗仪-第7周作业

PIL能够实现许多图片数据的底层表示和处理，比如在相素层面进行相关分析等，并支撑常用的图像检索任务。请围绕其相关功能，结合异常捕获和自定义异常，完成如下题目。

1. 异常捕获。实现ImageQuery类的create\_and\_image方法，其利用PIL.Image类的open方法打开并返回一个Image实例，但考虑到open方法可能产生FileNotFoundError或PIL.UnidentifiedImageError，请在该方法中对这两个异常进行捕获和处理（打印或记入日志，相关信息包括打开的文件路径和详细的异常描述）。

```
from PIL import Image
import imagehash
import scipy
import sys
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

class ImageExpection:
    def image_open():
        try:
            with
Image.open("C://Users//shiye//Desktop//homework6//beihang.jpg") as im1:
                c_hist_1 = im1.histogram()
                print(len(c_hist_1))
                ahash_1 = imagehash.average_hash(im1)
                print(ahash_1)

            with
Image.open("C://Users//shiye//Desktop//homework6//beihang2.jpg") as im2:
                c_hist_2 = im2.histogram()
                ahash_2 = imagehash.average_hash(im2)
                print(ahash_1-ahash_2)

        except FileNotFoundError: #捕获
            print("FileNotFoundError")
```

2. 图片的相似性计算。在ImageQuery类中实现一种简单图片相似性的计算方法pixel\_difference，即直接对两个图片逐相素相减，并累积求和差异的绝对值，继而除以相素总数。

**\*\*注意该方法可能会抛出一个叫ImageQueryShapeNotMatchError的自定义异常，其继承了ImageQueryError（本次作业自定义的顶层异常类），即当比较相似性的两张图片形状（长宽）不一致性时。请在该方法中抛出该异常，包含两个图片的形状信息。**

3. 图片的其他相似性计算。在ImageQuery类中实现更多的相似性计算方法。具体地，利用PIL.Image类的histogram方法，获取图片相素的直方图，进而用scipy.stats中的相关性计算方法来得到不同的相似性，如pearson, spearman, kendall等。

\*\*这些方法并不要求图片形状一致。注意，这些相似性方法还能够返回显著性。

```
p_cor = scipy.stats.pearsonr(c_hist_1, c_hist_2)#用scipy.stats中的相关性计算方法来得到不同的相似性
print(p_cor)

s_cor = scipy.stats.spearmanr(c_hist_1, c_hist_2)#用scipy.stats中的相关性计算方法来得到不同的相似性
print(s_cor)

k_cor = scipy.stats.kendalltau(c_hist_1, c_hist_2)#用scipy.stats中的相关性计算方法来得到不同的相似性
print(k_cor)

ImageExpection.image_open()
```

4. (附加) 安装imagehash库，利用其提供的一些hash算法（如average\_hash）等，来计算两张图片间的相似性，即其hash值的差异。
5. (附加) 实现ImageQuery类中的load\_images方法，借助\_creat\_an\_image，其能够加载并生成大量的Image实例，同时，还返回图片名称与其Image实例的对应。
6. (附加) 实现ImageQuery类中的search\_image方法，其能够对用户输入的一张图片（如通过路径输入），与5中加载的所有图片进行相似性计算（用参数指定计算哪种相似性）并排序，继而返回相似性小于某阈值（一个参数）的最相似的前k张图片，并利用matplotlib可视化出来。注意，该方法同样会产生异常，即ImageQueryNoSimilarImageFoundError,其同样继承ImageQueryError,并且在与所有图片的相似性过小（小于阈值）或均不显著（3中计算的相似性）时抛出。
7. (附加) 观察并比较不同相似性的计算速度和检索效果。同时，思考2, 3中的相似性计算能否进一步封装（比如类ImageSilarity，其中有一个方法叫similarity），进而与6中的search\_image进行更加抽象的依赖（只与抽象的ImageSimilarity类依赖，不与具体实现不同相似性计算的子类如ImagePixelDifferenceSimilarty, ImagePersonSimilarity等依赖）。

参考资料：

<https://pillow.readthedocs.io/en/stable/reference/Image.html#PIL.Image.Image.histogram>

<https://docs.scipy.org/doc/scipy/reference/stats.html>

<https://pypi.org/project/ImageHash/>

8. 完整代码：

```
from PIL import Image
import imagehash
import scipy
import sys
import cv2
import numpy as np
import matplotlib.pyplot as plt

# # 自定义异常类 MyError , 继承普通异常基类 Exception
```

```

# class ImageQueryShapeNotMatchError(ImageQueryError):
#     def __init__(self, shape):
#         self.shape = shape
#
#     def __str__(self):
#         return repr(self.shape)
# try:
#     num = input("请输入数字: ")
#     if not num.isdigit(): # 判断输入的是否是数字
#         raise ImageQueryShapeNotMatchError(num) # 输入的如果不是数字, 手动指定
#         抛出异常
# except ImageQueryShapeNotMatchError as e:
#     print("ImageQueryError", e.value)

class ImageExpection:
    def image_open():
        try:
            with
Image.open("C://Users//shiye//Desktop//homework6//beihang.jpg") as im1:
                c_hist_1 = im1.histogram()
                print(len(c_hist_1))
                ahash_1 = imagehash.average_hash(im1)
                print(ahash_1)

            with
Image.open("C://Users//shiye//Desktop//homework6//beihang2.jpg") as im2:
                c_hist_2 = im2.histogram()
                ahash_2 = imagehash.average_hash(im2)
                print(ahash_1-ahash_2)

        except FileNotFoundError:#捕获
            print("FileNotFoundError")

        p_cor = scipy.stats.pearsonr(c_hist_1, c_hist_2)#用scipy.states中的相
        关性计算方法来得到不同的相似性
        print(p_cor)

        s_cor = scipy.stats.spearmanr(c_hist_1, c_hist_2)#用scipy.states中的相
        关性计算方法来得到不同的相似性
        print(s_cor)

        k_cor = scipy.stats.kendalltau(c_hist_1, c_hist_2)#用scipy.states中的
        相关性计算方法来得到不同的相似性
        print(k_cor)

        try:
            plt.hist(im1.ravel(), 256, [0, 256])
            plt.show("直方图")
            color = ("blue", "green", "red")
            for i, color in enumerate(color):
                hist = cv2.calcHist([im1], [i], None, [256], [0, 256])
                plt.plot(hist, color=color)
                plt.xlim([0, 256])
            plt.show()

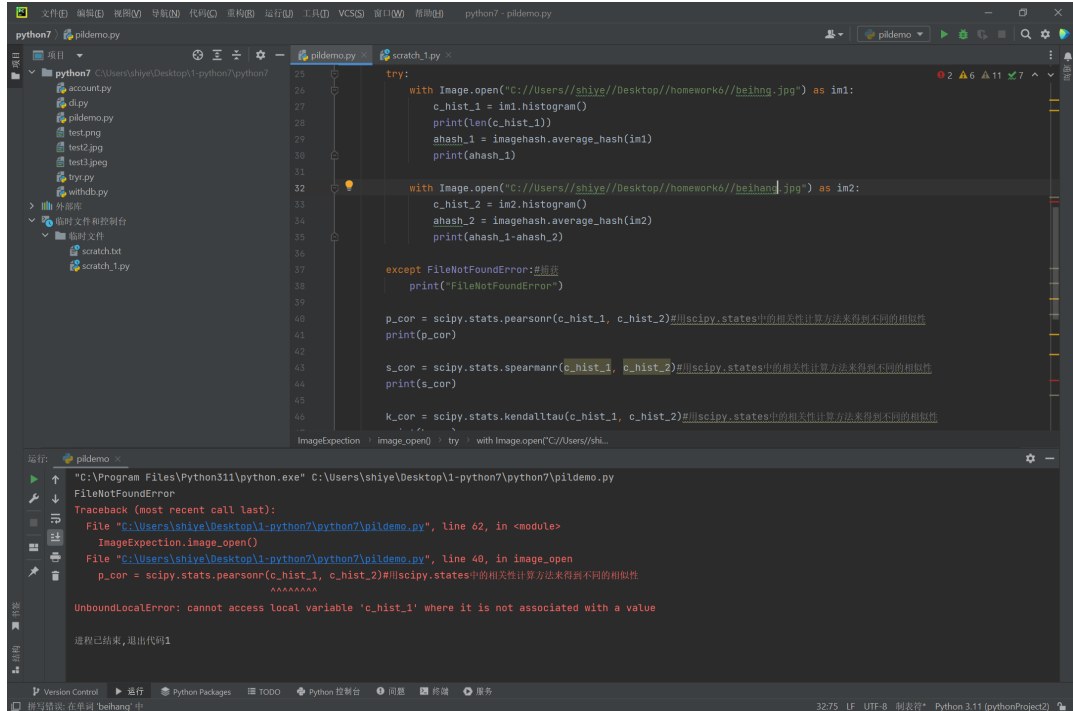
```

```
except ImageQueryShapeNotMatchError: # 捕获
    print("ImageQueryError")
```

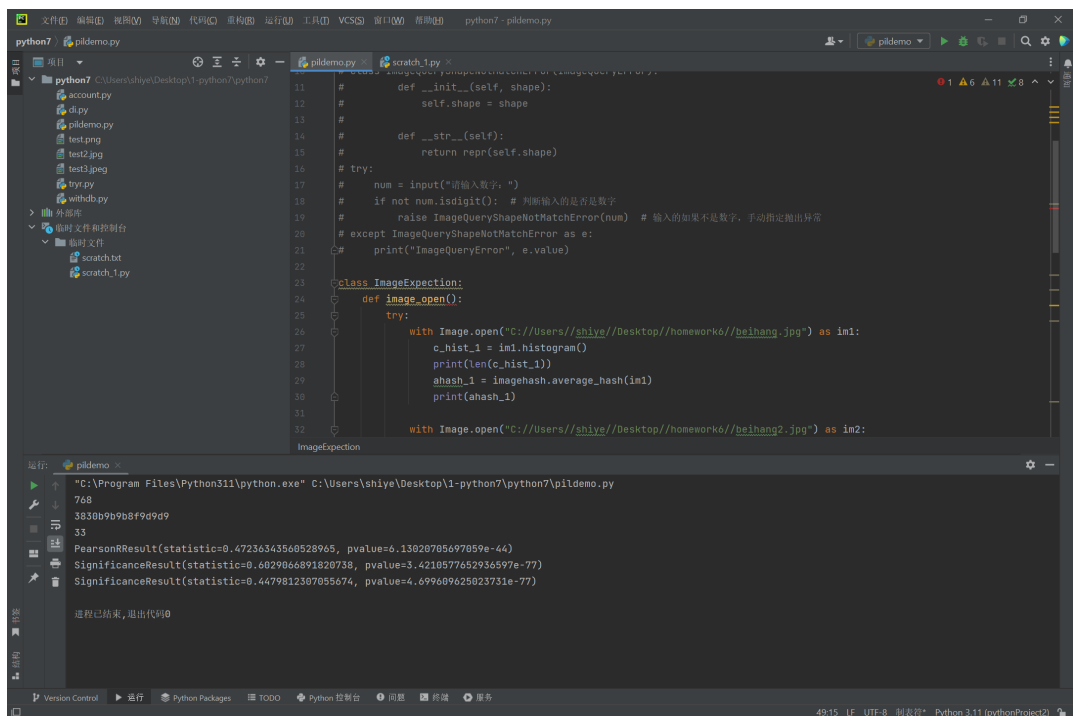
```
ImageExpection.image_open()
```

输出结果：

### 1. 照片出错捕捉：



### 2. 正确路径输出：



### 3.

