Otsu's Method

Author : 许泽资 5140379068

Introduction

- 整篇文档完成了两个任务。首先，实现了otsu算法；之后，利用otsu算法分开了cherry.png的前后景，选出了淤青。
- Otsu.py 是简单的实现，完成了分离前后景的工作。但是若要实现进一步的工作，比较难增加代码，十分难看，所
  之后摒弃不用，推荐使用其他两个文件。
- NewOstu.py 是新一代的Otsu包装，是专门对cherry.png 这个图片所做的特殊包装。OtsuCherry 类提供了一系
  函数，使得使用非常方便。
- BytesOtsu.py 为了更具通用性，ByteOtsu 类通过对bytes的操作进行处理，集中实现了Otsu方法，而对PIL模块
  再也没有依赖，成为独立而可配置的底层模块。OtsuCherry 类就是对该类的封装。
- 实现处理淤青的方法简述如下。首先找出前景，即cherry所在的部分；之后再进行一次Otsu's method, 但是背景不
  再参与运算。前景中较亮部分被定义为淤青部分，通过找到阈值，就可以方便的分出淤青和正常的部分了。

Python Environment

- python3
- pip install Pillow

Class Definition

- Otsu.py

```
class MyOtsuMethod:
    def __init__(self, file_path):
        '''
        初始化类，从file_path中读取图片，并转化为灰度图，
        再进行统计以及Otsu's method 分析的初始化。
        它只能获取处理后的binary图像。
```

```
        '''
        return

    def process_gray_level(self):
        '''
        开始Otsu's method 处理
        '''
        return

    def show(self):
        '''
        展现现在的图片
        '''
        return

    def saveTo(self, path):
        '''
        将处理好的图片储存到path的位置
        '''
        return

################ Demo ####################
def test_main():
    otsu = MyOtsuMethod('D:/cherry.png')
    otsu.process_gray_level()
    print(otsu.get_threshold())
    otsu.show()
    otsu.saveTo('D:/binaryCherry')
```

- NewOstu.py

```
class OtsuCherry:
    def __init__(self, path):
        '''
        初始化类，从file_path中读取图片，并转化为灰度图，
        再等待进一步处理。
        '''
        return

    def saveTo(self, path, type='B'):
        '''
        储存到path路径，再根据type来决定储存类型。
        type:
            'B' : 以binary的方式储存处理结果
            'F' : 以foreground取景的方式储存处理结果
            'BA': 以background取景的方式储存处理结果
        '''
        return
```

```
    def cutTheBruiseAndSave(self, path):
        '''
        搞定cherry上面的淤青。由于对淤青定义不明确，就把
        较为明亮的月牙部分当做淤青处理。
        处理完之后，直接存入path路径。
        '''
        return

################ Demo ###################
def test_main():
    otsu = OstuCherry('D:/cherry.png')
    otsu.saveTo('D:/back.png', type='BA')
    otsu.saveTo('D:/fore.png', type='F')
    otsu.cutTheBruiseAndSave('D:/bru.png')
```

- BytesOtsu.py

```
class ByteOtsu:
    """
    Class: Byte Otsu
        It takes in bytes stream, and output the result in
    user's demand.
        It provides get_binary()        # return the bytes of bin-image
                    get_foreground()    # make the background 0
                    get_background()    # make the foreground 255
                    get_anti_sigh()     # make the foreground 0, used in bruis

    Alert: Because we use the basic integer object, it can process not-so-
        big data. In this class, you are not allowed to change a image
        content.
    """

    def __init__(self, grayLevel, ignore = []):
        '''
        Constructor
            We get in some bytes data, and will pre-process on that. ignore is
        a list of 0~255, indicate which kind of pixel gray level is not allowe
        to be counted.
        :param grayLevel: bytes stream
        :param ignore: unwanted gray level sets
        '''
        return

    ###################################### fetch the data ###################

    def get_binary(self):
        '''
```

```
        获取binary处理结果
        '''
        return bytes()

    def get_foreground(self):
        '''
        获取前景，即低于阈值的部分直接置为0
        '''
        return bytes()

    def get_background(self):
        '''
        获取背景，即高于阈值的部分直接置为255
        '''
        return bytes()

    def get_anti_sigh(self):
        '''
        获取背景，但高于阈值的部分直接置为0
        主要用于处理淤青的时候使用。
        '''
        return bytes()

### demo part is used inside OtsuCherry class
```