第 3 讲　图像的清洗

3.1 图像的清洗

对于从网络下载的图片，有图像无法打开、有图像文件无效的情况.——需要清洗，删除这类文件。

去除无效文件、分离彩色图与灰度图，分离尺寸太小的文件、统一图像文件格式

3.1.1 图像基本知识

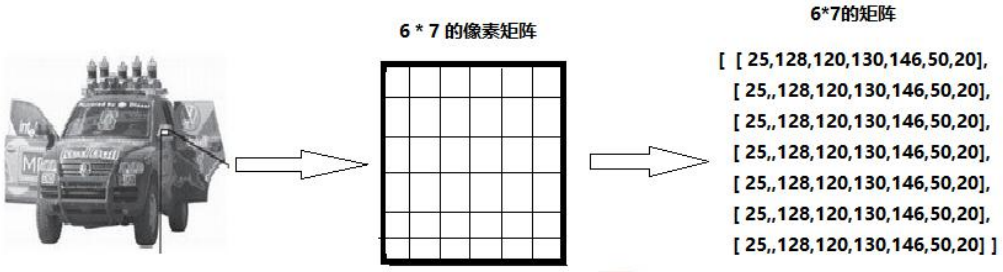
1.图像基础

（1）图像种类

二值图像（黑白）：每个像素只用 1 位来表示，取值为 0 和 1。

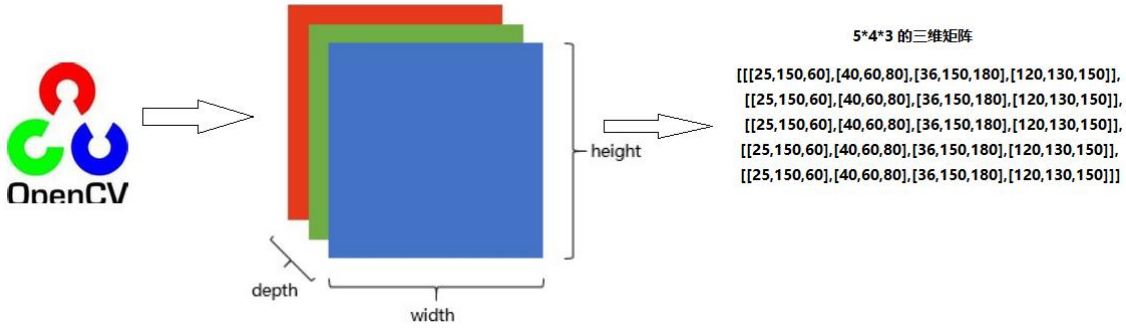
1	1	0	0	1	1
1	1	1	0	1	0
1	0	1	1	1	0
1	0	1	1	0	1

灰度图像：256 个灰度值，0~255，8 位的二进制值，每个像素用 8 位（1 个字节）表示。



彩色图像：RGB（R 红色通道，G 绿色通道，B 蓝色通道）每个像素有 3 个分量，需要有 3 个字节表示。

24*24*3byte



（2）图像的大小

图像的主要参数：分辨率、图像大小、图像的颜色

分辨率：图像的行列数目

图像的大小是指在磁盘上存储整幅图像所占用的字节数。可按下面的公式计算：

图像文件的字节数= 图像分辨率 ╳ 量化位数 / 8

例如：

一幅分辨率为 640 ╳480 的黑白图像，文件的大小为：

（ 640 ╳ 480 ）/ 8 = 38400（B）= 37.5(KB)

一幅同样分辨率的图像，图像深度为 8 位。则图像文件的大小为：

（ 640 ╳ 480 ）╳ 8/ 8 = 307200（B）/1024= 300(KB)

一幅同样大小的真彩色图像，则图像文件的大小为：

（ 640 ╳ 480 ）╳ 24/ 8 = 921600（B）= 900(KB)

2.图像基本操作 opencv

pip install opencv-python

（1）读取图像

image=cv2.imread(img_path,flag) 读取图片，返回图片对象

img_path: 图片的路径，即使路径错误也不会报错，但打印返回的图片对象为 None

flag：cv2.IMREAD_COLOR，读取彩色图片，图片透明性会被忽略，为默认参数，也可以传入 1

cv2.IMREAD_GRAYSCALE,按灰度模式读取图像，也可以传入 0

cv2.IMREAD_UNCHANGED,读取图像，保持原格式不变，也可以传入-1

```
import cv2
lena=cv2.imread('./testImages/lena.jpg',-1)
print(lena)
```

（2）显示图像

cv2.imshow(window_name,img)：显示图片，窗口自适应图片大小

window_name: 指定窗口的名字

img：显示的图片对象

可以指定多个窗口名称，显示多个图片

```
cv2.imshow('demo',lena)
```

cv2.waitKey()

cv2.waitKey(millseconds)　等待按键，用户键盘按键，语句执行，获得返回

millseconds：传入时间毫秒数，<0 或等于 0 时，会一直等待键盘事件

cv2.destroyAllWindows(window_name)

window_name：需要关闭的窗口名字，不传入时关闭所有窗口

```
cv2.destroyAllWindows()
cv2.imwrite('./testImages/lena1.jpg',lena)
```

（3）查看图像属性

shape：获取图像的形状，返回行数、列数、通道数的元组（行，列，通道 ）

size：返回图像的像素点个数

dtype：返回图像的数据类型

ndim:3

(H,W,C)=img.shpe()

（4）图像保存

cv2.imwrite(file, img, num)

file:要保存的文件名

img：要保存的图像

num：不同格式文件，可选

边学边练：

工作路径的 data 目录下有一系列图片，通过编程实现文件清理。

先去除无法正常打开的文件，包括动图、不完整图片，

去除灰度图像，

去除尺寸小于 200*200 像素的图像，

把保留的图片转换成 jpg 格式，按顺序 001，,003 命名。

3.2 numpy 库

3.2.1 数组

（1）一维数组

一维数据由对等关系的有序或无序数据构成，采用线性方式组织

3.1413, 3.1398, 3.1404, 3.1401, 3.1349, 3.1376

列表和数组的区别：

列表：数据类型可以不同　3.1413，‘pi’，3.1404，[3.1401，3.1349]，‘3.1302’

数组：数据类型相同　【3.1413, 3.1398, 3.1404,】

（2）二维数组

二维数据由多个一维数据构成，是一维数据的组合形式

表格是典型的二维数组。

（3）多维数组维度
多维数据由一维或二维数据在新维度上扩展形成

3.2.2 numpy 创建数组

（1）ndarray 对象用于存放同类型元素的多维数组，通过 np.array 对象来创建。
arr1 = np.array([1, 2, 3, 4]) #创建一维数组
arr2 = np.array([[1, 2, 3, 4],[4, 5, 6, 7], [7, 8, 9, 10]]) #创建二维数组 3*4
arr3=
np.array([[[1,2,3,4],[4,5,6,7],[7,8,9,10]],[[11,12,13,14],[14,15,16,17],[17,18,19,20]],[[21,22,23,24],[24,25,26,27],[27,28,29,30]]])#创建三维数组 3*3*4

$$\begin{bmatrix} [1,2,3,4] \\ [4,5,6,7] \\ [7,8,9,10] \end{bmatrix}$$ 3*4 $$\begin{bmatrix} [[1, 2, 3, 4],[4, 5, 6, 7], [7, 8, 9, 10]] \\ [[11, 12, 13, 14],[14, 15, 16, 17], [17, 18, 19, 20]] \\ [[21,2 2, 23,2 4],[24, 25,2 6, 27], [27,2 8, 29,30]] \end{bmatrix}$$

（2）numpy 包含的数据类型
bool：布尔类型，True 或者 False
inti：长度取决于平台的数据类型，一般是 int32 或者 int64
int8：字节长度的整数
int16、int32、int64：16 位、32 位、64 位长度的整数
unit8、unit16、unit32、unit64：8 位...无符号整数
float16、float32、float64:16 位（1 位符号位、5 位指数位、10 位尾数）半精度浮点数
complex64、complex128：复数，实部和虚部都是 32/64 位浮点数。
（3）ndarray 对象常用属性

属性	说明
ndim	返回 int。表示数组的维数
shape	返回 tuple。表示数组的尺寸，对于 n 行 m 列的矩阵，形状为(n,m)
size	返回 int。表示数组的元素总数，等于数组形状的乘积
itemsize	返回 int。表示数组的每个元素的大小（以字节为单位）
dtype	返回 data-type。描述数组中元素的类型
T	返回转置数组
flat	数组的以为迭代器

3.2.3 numpy 文件常用操作

（1）np 的保存
NumPy 可以用专有的二进制类型保存数据，文件后缀为 npy。通过 np.load 和 np.save

这两个函数可以方便的读写数组文件，自动处理元素类型和 shape 等信息。

np.save()

```
>>> import numpy as np
>>> a =np.array([[ 1, 2, 3, 4], [ 5, 6, 7, 8], [ 9, 10, 11,12]],dtype=np.float32)
>>> np.save("a.npy",a)
>>> new_a = np.load("a.npy")
>>> new_aarray([[1.,2.,3.,4.], [5., 6., 7., 8.],[9., 10., 11., 12.]], dtype=float32)
```

（2）一维序列数组

np.arange(start, end, step)：等差数列的一数组

（3）等差数列

np.linspace(strat, end, count)

（4）等比数列

np.logspace(start, end, count)

（5）全 0 或 1 或全 X 数组

np.zeros(shape, dtype)

np.ones(shape, dtype)

np.full(shape, X)

（6）数组维度的变换

a.reshape(newshape):改变数组的维度，新数组元素数量与原数组一样，返回新数组

a.resize(newsize)：改变数组维度，直接修改原始数组

3.2.4 图像文件 numpy 存取

读取一幅图片，显示图像的形状、像素数目及数据类型。示例代码如下：

```
>>> import cv2
>>> img = cv2.imread("lena.jpg")
>>>print(img)
>>> print("图像形状:",img.shape)
>>> print("像素数目:",img.size)
>>> print("数据类型:",img.dtype)
>>>np.save("a.npy",img)     #将图像存到 numpy 文件中
>>>a=np.load('lena.npy')#读入 numpy 文件
>>>cv2.imwrite('./testImages/lena2.jpg',a)#将 numpy 文件还原为图像
```