

2022-2023学年度第一学期课程

Python与数据科学

Python and Data Analysis

新疆农业大学计算机与信息工程学院
cs.xjau.edu.cn



孟小艳
AI系主任
副教授，博士
13609979825
mxy@xjau.edu.cn

| 第8周 | 专题 | 章节 | 学时 | 负责老师 | 实验设置 |
|-------|-----------|--------|----|------|------|
| 7-8节 | 不同类型数据的处理 | OpenCV | 2 | 孟小艳 | 实验四 |
| 9-10节 | | 时间序列 | 2 | | |

第8章 OpenCV图像处理

目录

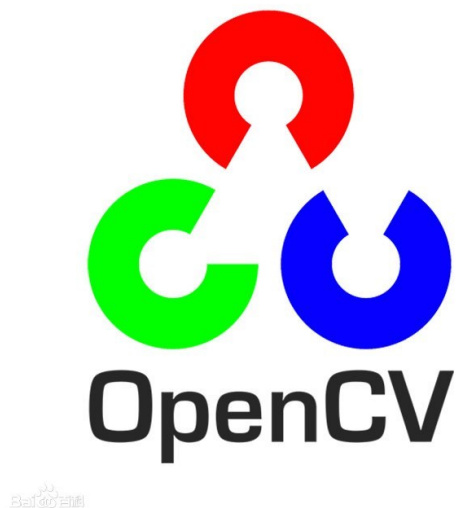
8.1 OpenCV介绍

8.2 图像基础知识

8.3 图像的存储与读取

8.4 图像的操作处理

8.1 OpenCV介绍



Open Source Computer Vision Library

是一个跨平台、开源、提供多语言接口的计算机视觉库，主要用于图像处理。

<https://baike.baidu.com/item/opencv/10320623?fr=aladdin>



8.1 OpenCV介绍

➤ OpenCV Python的下载、安装与导入

(1) 下载并安装Python

(2) 安装依赖库 (numpy、matplotlib)

(3) 下载并安装opencv_python

(4) 导入 `import cv2 as cv`

目录

8.1 OpenCV介绍

8.2 图像基础知识

8.3 图像的存储与读取

8.4 图像的操作处理

8.2 图像基础知识

➤ 计算机成像的原理

数字图像的采集是利用大量的光敏传感器构成的阵列来获取图像。

图像数字化就是把一副画面的数据转换为计算机能够处理的数字形式。



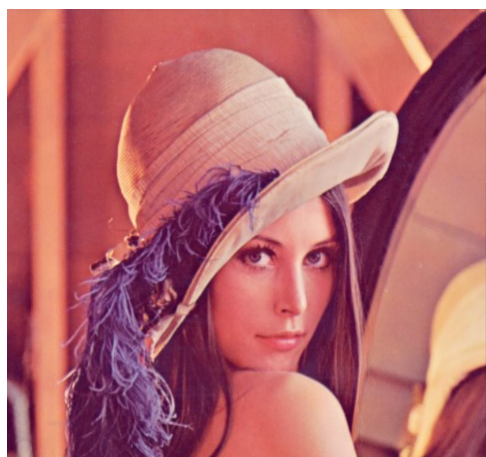
8.2 图像基础知识

➤ 采样和量化

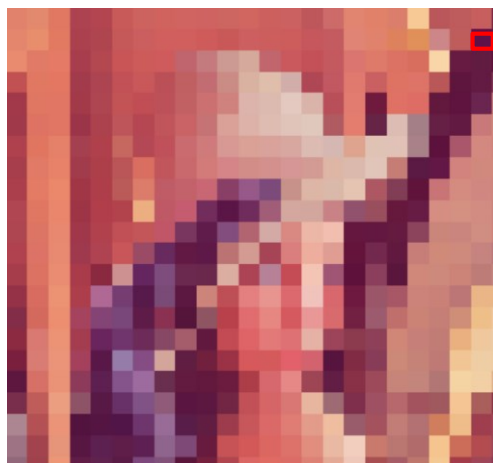
采样（Sampling）——把一幅连续图像在空间上分割成 $M \times N$ 个网格，每个网格用一亮度值来表示。
一个网格称为一个像素。

量化（quantization）——把采样点上对应的亮度连续变化区间转换为单个特定数码的过程。

量化后，图像就被表示成一个整数矩阵。

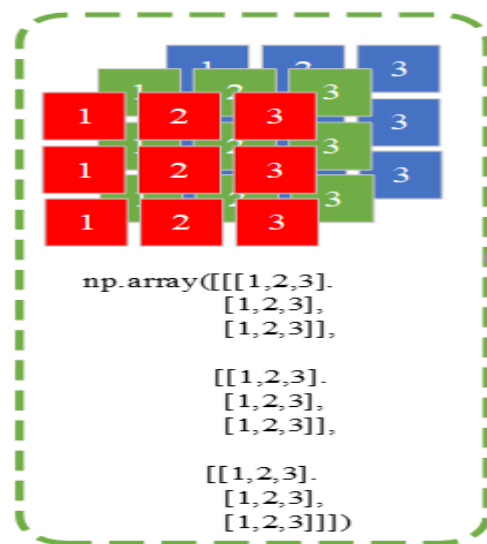


采样



像素

量化

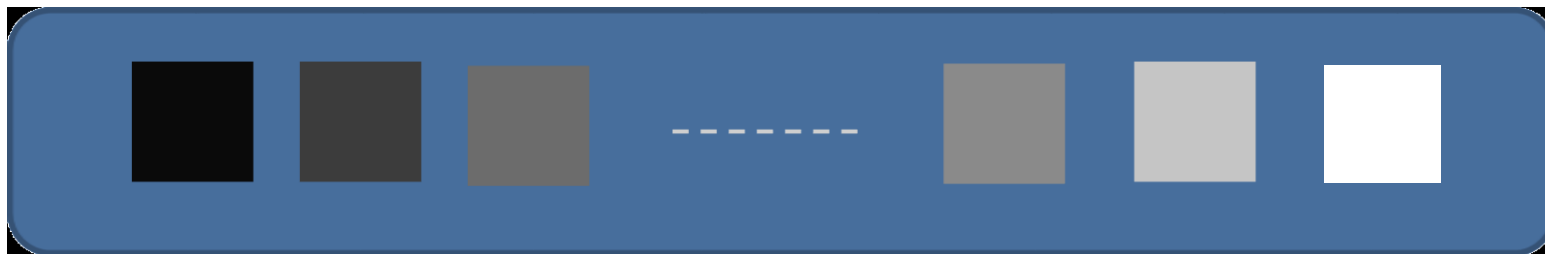


8.2 图像基础知识

➤ 灰度级

表示每个采样点在计算机中存储的灰度的级数。用2的整数次幂

从最暗到最亮
一共有多少个
层级的灰度？



计算机通常使用256个灰度级，即8位二进制表示图像的灰度级。

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|



黑色

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|



白色

例如：一副8位的图像，表示每个采样点都有 $2^8=256$ 级

8.2 图像基础知识

灰度级也叫量化等级，灰度级越多，图像层次越丰富，视觉效果就越好。



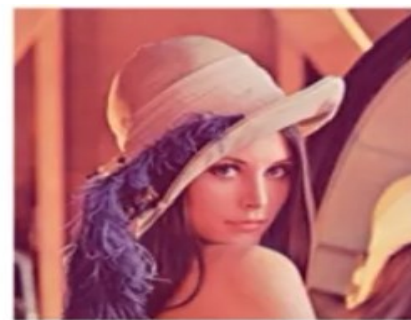
256级---→ 4级

8.2 图像基础知识

➤ 图像分类

按**颜色**和**灰度**的多少进行分类

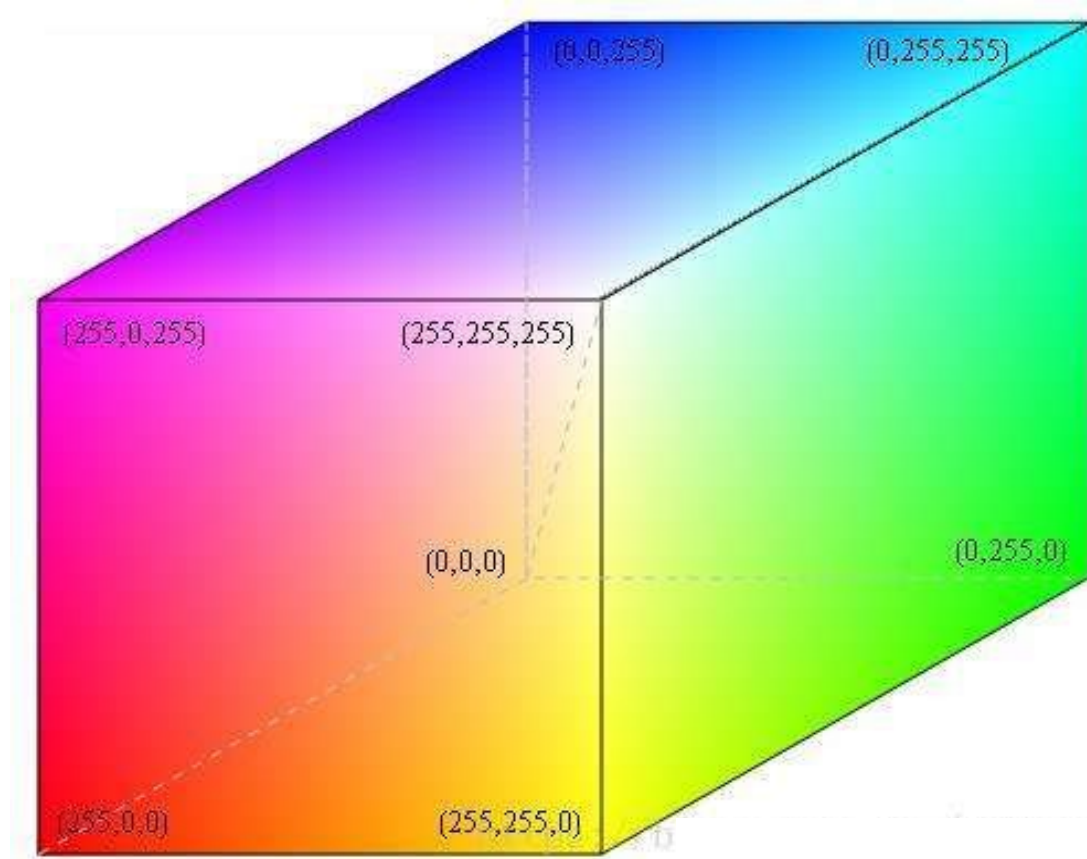
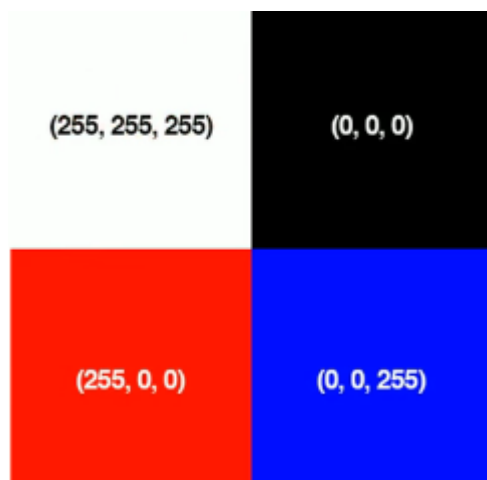
- (1) 二值图像——图像构成的每一像素点，仅有0，1两种取值。0为黑色，1为白色。
- (2) 灰度图像——每个像素只有一个采样颜色的图像。
(颜色是相同的，只是灰度级别不同而已，类似于绿、浅绿、深绿、墨绿)
- (3) RGB彩色图像——每个像素的颜色值都由RGB三原色来表示，直接存放在矩阵中。
- (4) 索引图像——是一种把像素值直接作为RGB调色板下标的图像。



8.2 图像基础知识

彩色图像有三个通道 R, G, B

R 红色
G 绿色
B 蓝色
取值范围0~255



注意：OpenCV中以BGR形式存储

目录

8.1 OpenCV介绍

8.2 图像基础知识

8.3 图像的存储与读取

8.4 图像的操作处理

8.3 图像的存储与读取

8.3.1 窗口操作（创建、显示、释放）

8.3.2 图片操作（加载、显示、保存）

8.3.3 视频操作（采集、播放、录制）

8.3.1 窗口操作——创建

`namedWindow (winname , flags)`

- ◆ 功能:新建一个显示窗口
- ◆ 参数1: 新建窗口的名称, 是字符串;
- ◆ 参数2: 窗口类型, 默认为 `WINDOW_AUTOSIZE`
 - `WINDOW_AUTOSIZE` 窗口大小自动适应图片大小, 不可手动更改。
 - `WINDOW_NORMAL` 窗口大小可以随意拖动改变大小

8.3.1 窗口操作——显示

`imshow (winname, mat)`

- ◆ 功能:在指定的窗口中显示一个图像
- ◆ 如果没有创建窗口，会默认创建一个AUTOSIZE的窗口
- ◆ 参数1：窗口的名称，是字符串；
- ◆ 参数2：矩阵

8.3.1 窗口操作——关闭

```
destroyWindow ()  
destroyAllWindows ()
```

- ◆ 功能:关闭窗口，同时释放与窗口相关的内存空间

8.3.1 窗口操作

窗口操作

- ◆ **namedWindow()**
- ◆ **imshow()**
- ◆ **destroyAllWindows()**
- ◆ **waitKey()**
- ◆ **resizeWindow()**

例8-1 窗口操作

8.3.2 图片操作——加载

`imread (picname , flag)`

- ◆ 功能:在窗口中加载图片
- ◆ 有返回值，返回值一个mat
- ◆ 参数1：窗口的名称，是字符串；
- ◆ 参数2：图像类型转换
 - 1,：IMREAD_COLOR，常规显示
 - 0：IMREAD_GRAYSCALE，将加载的图片转换为灰度图像

8.3.2 图片操作——显示

```
imshow ( picname , mat)
```

- ◆ 功能:在窗口中加载图片
- ◆ 参数1: 图片的名称, 要带路径, 是字符串;
- ◆ 参数2: 加载的图片对象的矩阵, 从imread () 获取

8.3.2 图片操作——保存

```
imwrite (name, img)
```

- ◆ 功能:：保存图片
- ◆ 参数1:：要保存的文件名，可以带路径和扩展名，是字符串；
- ◆ 参数2:：是mat类型

8.3.2 图片操作

图片操作

- ◆ **imread()**
- ◆ **imshow()**
- ◆ **imwrite()**

例8-2, 8-3 图片操作

Python中读取图像的多种方法:

◆ opencv

◆ matplotlib

◆ pillow

注意:

1. Pillow读入的图像是一个对象文件，需要用array()方法做图片转矩阵的操作.
2. OpenCV读入是numpy矩阵，但彩色图片是以BGR顺序存储
3. matplotlib读入的也是numpy矩阵，是RGB顺序

8.3.3 视频操作——采集

VideoCapture(index)

read()

release()

- ◆ VideoCapture(index) 功能: 建立一个虚拟采集器, 打开摄像头设备;
index是数字, 默认摄像头序号为0;
- ◆ read() : 读取视频帧, 不带参数, 有两个返回值;
 - 第一个返回值是状态值, 读取到返回true, 否则false
 - 第二个返回值是视频帧
- ◆ release() 功能: 释放视频占用设备资源

8.3.3 视频操作——播放视频文件

VideoCapture(filename)

- ◆ VideoCapture(filename) 功能: 从视频文件中读取视频帧

8.3.3 视频操作——视频录制

```
VideoWriter(filename, fourcc,fps,framesize)  
write()  
release()
```

- ◆ VideoWriter () 功能:从视频文件中读取视频帧
- ◆ 参数1: 输出文件
- ◆ 参数2: 多媒体文件格式, 是编码器格式
- ◆ 参数3: 帧率
- ◆ 参数4: 分辨率

8.3.3 视频操作

视频操作

◆ **VideoCapture()**

◆ **VideoWriter()**

◆ **read()**

◆ **write()**

◆ **release()**

例8-4 视频操作

8.3 图像的存储与读取

8.3.1 窗口操作（创建、显示、释放）

8.3.2 图片操作（加载、显示、保存）

8.3.3 视频操作（采集、播放、录制）

8.3 图像的存储与读取

窗口操作

- ◆ `namedWindow()`
- ◆ `imshow()`
- ◆ `destroyAllWindows()`
- ◆ `waitKey()`
- ◆ `resizeWindow()`

图片操作

- ◆ `imread()`
- ◆ `imshow()`
- ◆ `imwrite()`

视频操作

- ◆ `VideoCapture()`
- ◆ `VideoWriter()`
- ◆ `read()`
- ◆ `write()`
- ◆ `release()`

目录

8.1 OpenCV介绍

8.2 图像基础知识

8.3 图像的存储与读取

8.4 图像的操作处理

8.4 图像的操作处理

8.4.1 图像色彩空间转换

8.4.2 图像像素操作（读写、算数）

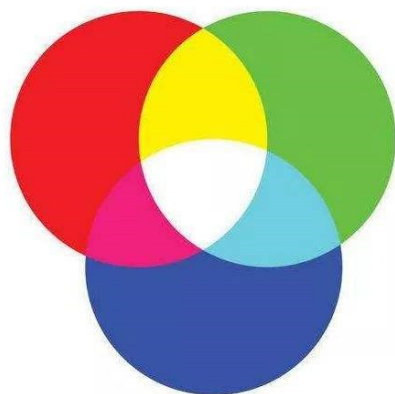
8.4.3 利用OpenCV绘制图像

8.4.4 图像变换

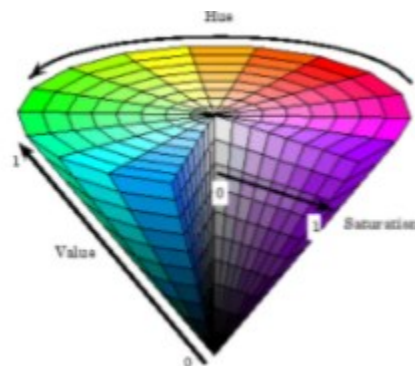
8.4.1 图像色彩空间转换

◆ 几种颜色空间

RGB、HLS、HSV、HSB、YCrCb、CIE XYZ、CIE Lab

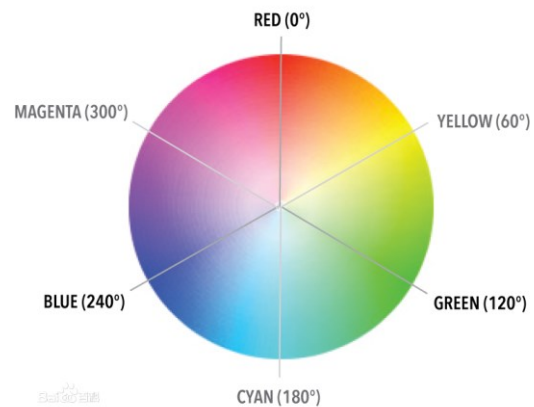


RGB



HSV

色调H (0-360)
饱和度S(0%-100%)
明度V(0-100)



HSB

H代表色相（色度）
S代表饱和度（0%-100%）
B代表亮度（0° -360°）

8.4.1 图像色彩空间转换

◆ 几种颜色空间

RGB、HLS、HSV、HSB、YCrCb、CIE XYZ、CIE Lab

```
cvtColor (img, code[,dst[, dstCn]])
```

◆ 功能：将图像从一个颜色空间转换为另一个颜色空间。

◆ 参数1：原图像

◆ 参数2：需要进行色彩空间转换的结果。

0对应COLOR_BGR2BGRA, 1对应 COLOR_BGRA2BG.....

例8-5 色彩空间转换

8.4.2 图像像素操作

◆ 几种像素操作

加法、减法、乘法、除法、位运算（与、或、非、或、异或）

`add (A, B)`

`Subtract(A, B)`

- ◆ `add`: 增加亮度, 变亮;
- ◆ `subtract`: 降低亮度, 变暗
- ◆ `multiply`: 亮的更多一些;

例8-6 加法

8.4.2 图像像素操作

◆ 图像融合

```
addweighted(A, alpha,B ,beta,gamma)
```

- ◆ 功能：将2副图像融合，生成一张新图片。
- ◆ `alpha`：图A占比
- ◆ `bata`：图B占比
- ◆ `gamma`：静态权重，生成新图片中所有元素都加这个权重

注意：融合的两张图片需要具有同样的h,w,c

OpenCV中处处皆
Numpy

OpenCV中用到的矩阵都需要转换成Numpy数组

需要掌握的Numpy操作有：

◆ 创建矩阵

(`array()`，全零数组`zeros()`，全1数组`ones()`，全值数组`full()`，单元数组`identity()`)

◆ 检索与赋值

◆ 截取子数组

8.4.3 利用OpenCV绘制图像

◆ 可以绘制的图形

线、圆形、矩形、椭圆、多边形、文本

```
line(img, startpt, endpt,color, thickness,lineType)
```

```
circle(img,startpos,r,color)
```

```
rectangle(img, pos1, pos2, color, thickness)
```

```
ellipse(img,中心点,长宽的一半, 角度, 开始角度)
```

```
Polylines(img,pts,isclosed,color,)
```

8.4.3 利用OpenCV绘制图像

◆ 鼠标控制

例8-7 绘制图像

8.4.4 图像变换

➤ 色彩处理

- 灰度化
- 二值化
- 色彩提取
- 直方图均衡
- 亮度、饱和度、色调调整

8.4.4 图像变换

➤ 形态处理

- 旋转、平移
- 缩放、剪裁
- 图像相加、相减
- 透视变化
- 图像膨胀

8.4.4 图像变换——缩放

`resize (scr, dsize, fx,fy,interpolation)`

- ◆ 功能：改变图片大小。
- ◆ 参数1scr：原始图片；
- ◆ 参数2dsize：目标大小；
- ◆ 参数3、4fx, fy：横纵坐标的缩放因子（注意：dsize不能同时设置）
- ◆ 参数5：缩放算法（有临近差值法、双线性插值、三次插值、区域插值）

例8-8 缩放

OpenCV-Python教程简介

http://www.cvtutorials.com/#/opencv_py4.5.3/cvpy4.5.3_c0

本章结束