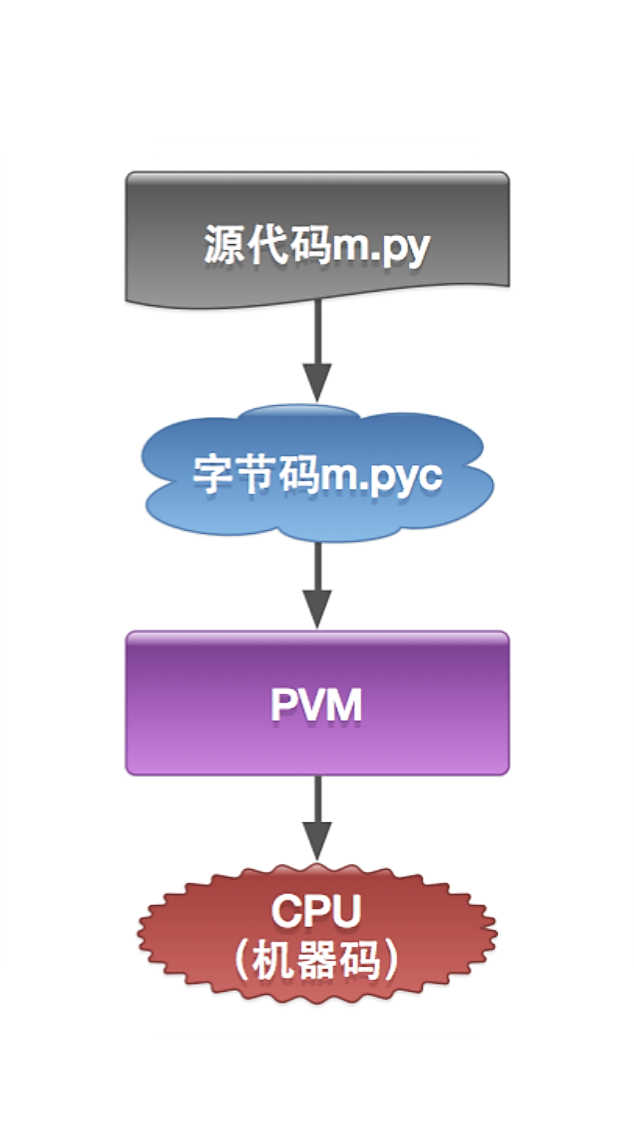
实验二 Python及外部库安装与使用

1. C vs Python
   1. C语言是编译语言，Python是脚本语言，相同代码下，C语言的运行比python快



1. Python运行执行过程
   1. 数组和List

C语言的数组用于存储数字，List中可以包含多种类型的数据

代码：

# C

int a[10]

# Python

list1 = ['physics', 'chemistry', 1997, 2000]

* 1. include和import

C语言中使用库函数，需要用include引入头文件；而在python中引入别的模块或者函数时，需要用import引入。

两者的不同机制是，C语言中include是告诉预处理器，这个include指定的文件的内容，要当作本地源文件出现过；而python中的通过简单的import 导入，或者是 import \*\*\* as \*\*。

代码：

# C

include<math.h>

# Python

import numpy

import numpy as np

* 1. 全局变量

C语言中声明全局变量，如果值是恒定的，那么用#define声明，如果是声明全局且变量值可变，那么直接int a；在python中，声明全局变量时，需要加上global，在函数里面使用时需先声明global \* ，若直接用该变量，那么python会重新创建一个新的本地对象并将新的值赋值给他，原来的全局变量的值并不变化。

代码：

# C

#define M\_PI 3.14159

# Python

global a

a = 3

def func():

#global a #print🡪4

a = 4

print(a)

func()

print(a)

#运行结果：

#3

1. Python2 vs Python3

由于Python2在2020年会停止维护更新，故本课程使用Python3进行学习。

* 1. print功能

在Python2中，print是一条语句，而Python3中作为函数存在。

代码：

# Python2

>>> print(”hello”, ”world”) #等价print (”hello”, ”world”)

(”hello”, ”world”)

# Python3

>>> print(”hello”, ”world”)

Hello world

* 1. 编码

Python2的默认编码器是ascii（Python诞生时还并未出现Unicode， Python2沿用了该方案），有关中文的程序中必须在顶部写# coding=utf-8；而Python3默认采用UTF-8作为编码，因此，无需再文件顶部写# coding=utf-8。

代码：

# Python2

>>> sys.getdefaultencoding()

’ascii’

# Python3

>>> sys.getdefaultencoding()

’utf-8’

* 1. True和False

在Python2中，True和False是全局变量，在数值上分别对应1和0，可以被重新赋值。Python3中修正了这个缺陷，True和False被设定为两个关键字，不允许被重新赋值。

代码：

# Python2

>>> True = false

>>> True

False

# Python3

>>> True = 1

File ”<stdin>”, line 1

syntaxError: can’t assign to keyword

1. 外部库
   1. 外部库的安装

外部库的安装共有三种方法

第一种：直接将外部库整个放入python\Lib\site\_packages\下面

第二种：使用pip安装外部库 pip install \*\* (\*\*为库的名称)

第三种：pip install D:\matplotlib-2.2.4-cp27-cp27m-win\_amd64.whl

轮子下载网址：①https://pypi.org/；②http://mirrors.aliyun.com/pypi/simple/ （阿里云镜像）。

* 1. opencv-python库

opencv是一个非常强大的图像处理、分析库。cv指计算机视觉（computer version），它在图像处理方面应用广泛，包括目标检测，图像分割，人脸识别等多种功能。

在本次实验中主要讲解部分操作，感兴趣的同学可以访问以下网址，查询更多功能：<https://docs.opencv.org/4.4.0/d2/d96/tutorial_py_table_of_contents_imgproc.html>

在python中使用需要先导入该库：import cv2。

* + 1. 图像读写存

1. 读取图像

代码：

#全英文字母

img = cv2.imread('读取的文件路径', cv2.IMREAD\_GRAYSCALE)

#包含中文 需导入numpy

img2 = cv2.imdecode(np.fromfile('读取的文件路径',dtype=np.uint8), cv2.IMREAD\_GRAYSCALE)

函数参数为：读取的文件路径、以何种方式显示（如灰度，彩色等），对应标识如下：

enum {

IMREAD\_COLOR = 1, #未改变深度、颜色，默认值

IMREAD\_GRAYSCALE = 0, #8位，灰度图

IMREAD\_UNCHANGED = -1, #8位，彩色或非彩色

IMREAD\_ANYDEPTH = 2, #任意深度，灰度图

IMREAD\_ANYCOLOR = 4, #未改变深度，任意颜色

IMREAD\_LOAD\_GDAL = 8 #使用 gdaL 驱动程序

}

imread函数值能读取英文路径的图像，imdecode函数可读取含中文路径的图像。

1. 显示图像

代码：

#设置窗口是否可变

cv2.namedWindow('输出的窗口名', 0)

#在窗口显示图像

cv2.imshow('输出的窗口名', img)

#设置窗口打开时长

cv2.waitKey(0)

namedWindow函数参数为：输出的窗口名称（英文）、窗口显示方式：

enum{

WINDOW\_NORMAL = 0 #窗口大小可变

WINDOW\_AUTOSIZE #或省略，窗口大小不可变

}

imshow函数参数为：输出的窗口名称（英文）、要输出的图像。

waitKey函数参数为：显示窗口的时间长短，单位为ms。参数>0时，那么超过指定的时间返回-1；参数等于0时，一直显示窗口直到有按键按下，返回按键的值（多用于视频和摄像头的显示）。

1. 保存图像

代码：

#全英文字母

cv2.imwrite('输出文件名', img)

#包含中文

cv2.imencode('.jpg', img)[1].tofile('输出文件名')

imwrite函数参数为：输出的文件名、要输出的图像。

imencode函数参数为：输出的文件类型、要输出的图像、输出的文件名。

* + 1. 图像翻转

图像翻转是一个非常酷炫的小技能。就像手机的前置镜头，可以实时的镜面成像，这个功能利用cv2中的函数可以简单实现。

代码：

img\_flip = cv2.flip(img, flipCode)

第1个参数为待转换的图像，第2个参数为以何种方式转换，必须是整数：> 0：绕y轴旋转；0：绕x轴旋转；< 0：绕x, y旋转。

代码：

import cv2 #导入模块

img = cv2.imread('Lena.jpg', 1) #读取图像

cv2.imshow("Lena", img)

#水平翻转

h\_img = cv2.flip(img, 1)

cv2.imshow('Lena\_himg', h\_img)

#垂直翻转

v\_img = cv2.flip(img, 0)

cv2.imshow('Lena\_vimg', v\_img)

#水平加垂直翻转

hv\_img = cv2.flip(img, -1)

cv2.imshow('Lena\_hvimg', hv\_img)

cv2.waitKey(0) #等待事件触发，参数0表示永久等待

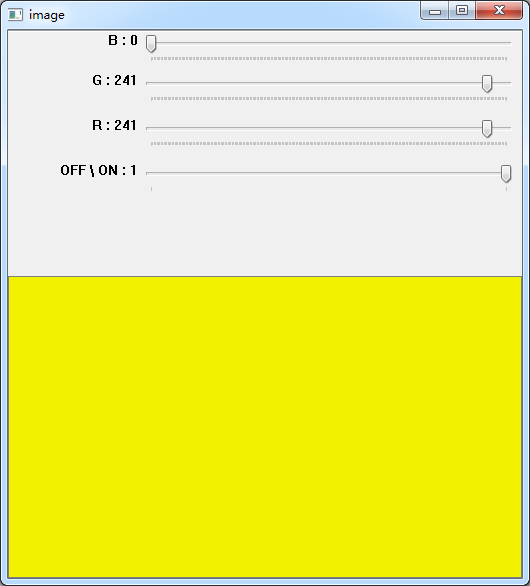
cv2.destroyAllWindows() #释放窗口

1. Lena图与翻转变换后的图像
   * 1. 自制调色板

这一小节通过设定滑动的轨迹条，设计一个自定义的调色板。目的是让大家了解函数的调用关系，熟悉python的语法规则，同时也了解一下opencv中对多波段图像的操作。



1. 自定义调色板

代码：

#创建一个轨迹条

cv2.createTrackbar('B ', 'image', 0, 255, nothing)

#获得轨迹条的值

b = cv2.getTrackbarPos('B ', 'image')

createTrackbar函数参数为：轨迹条名称、轨迹条所在窗口名称、滑块的初始位置、最大值、回调参数、用户传给回调函数的数据值（可选）。

getTrackbarPos函数参数为：轨迹条名称、轨迹条所在窗口名称

代码：

#自定义一个调色板

import cv2

import numpy as np

def nothing(x):

pass

#创建一个初始图像

img = np.zeros((300, 512, 3), np.uint8)

#创建一个名为image的窗口

cv2.namedWindow('image')

#创建四个轨迹条

cv2.createTrackbar('B ', 'image', 0, 255, nothing)

cv2.createTrackbar('G ', 'image', 0, 255, nothing)

cv2.createTrackbar('R ', 'image', 0, 255, nothing)

cv2.createTrackbar('OFF \ ON ', 'image', 0, 1, nothing)

while(True):

cv2.imshow('image', img)

k = cv2.waitKey(1)

if(k == ord(' ')): #获取从键盘输入的消息，即按空格键退出

break

#返回轨迹条所在值

b = cv2.getTrackbarPos('B ', 'image')

g = cv2.getTrackbarPos('G ', 'image')

r = cv2.getTrackbarPos('R ', 'image')

s = cv2.getTrackbarPos('OFF \ ON ', 'image')

#切记，名称一定要与上面定义名称相同，空格也需要记录

#显示三波段合成结果

if s == 0:

img[ : ] = 0

else:

img[ : ] = [b, g, r]

cv2.destroyAllWindows()

* 1. Matplotlib库

Matplotlib是Python的绘图库，其中的pyplot包封装了很多画图的函数。Matplotlib.pyplot中的函数会对当前的图像进行一些修改，例如：产生新的图像，在图像中产生新的绘图区域，在绘图区域中画线，给绘图加上标记等。Matplotlib.pyplot会自动记住当前的图像和绘图区域，因此这些函数会直接作用在当前的图像上。

在python中使用需要先导入该库：import matplotlib.pyplot as plt

* + 1. 绘图

代码：

#定义数据及线形

plt.plot([1,2,3,5], [5,3,2,1], 'r--', label='line', linewidth=1)

#显示图例

plt.legend()

#定义标题

plt.title('Title')

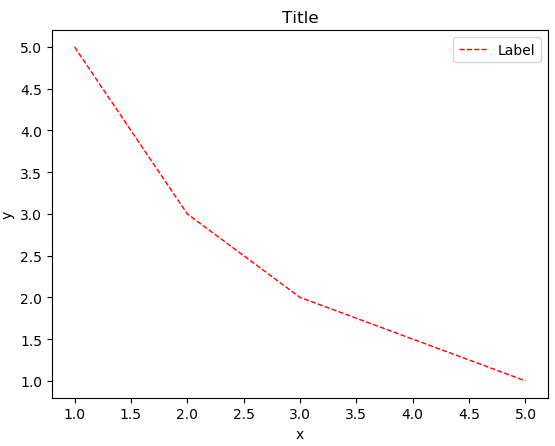
#定义x/y轴标签

plt.xlabel('x')

plt.ylabel('y')

#显示

plt.show()



1. 绘图样图

plot函数参数为：x轴数据（可省略，默认为0~n-1的自然数）、y轴数据（不可省略）、绘制格式（如下表所示）、数据标签、线宽度。

#表示颜色的字符

enum{

'b' #蓝色

'g' #绿色

'r' #红色

'c' #青色

'y' #黄色

'k' #黑色

}

#表示类型的字符

enum{

'-' #实线

'--' #虚线

'-.' #虚点线

':' #点线

'v' #下三角

'^' #上三角

}

* + 1. 图像统计（直方图）

将能看懂直方图作为数字图像处理三大基本功之一，充分说明了直方图在图像处理领域的重要程度，本节学习使用opencv中的直方图函数绘制直方图。

代码：

hist = cv2.calcHist([img], channels, mask, histSize, ranges [, …])

第1个参数为待计算的图像，需要写在方括号中，第2个参数为使用图像的第几个波段（灰度图像默认0），第三个参数为掩膜区域，第四个为目标区域直方图的量级值（对于8位图像默认256，写在方括号中），第五个为直方图的范围。

代码：

import cv2

import matplotlib.pyplot as plt

#读取目标图像并绘制直方图

#BGR

histb = cv2.calcHist([img], [0], None, [256], [0, 255])

histg = cv2.calcHist([img], [1], None, [256], [0, 255])

histr = cv2.calcHist([img], [2], None, [256], [0, 255])

cv2.imshow("02image", img)

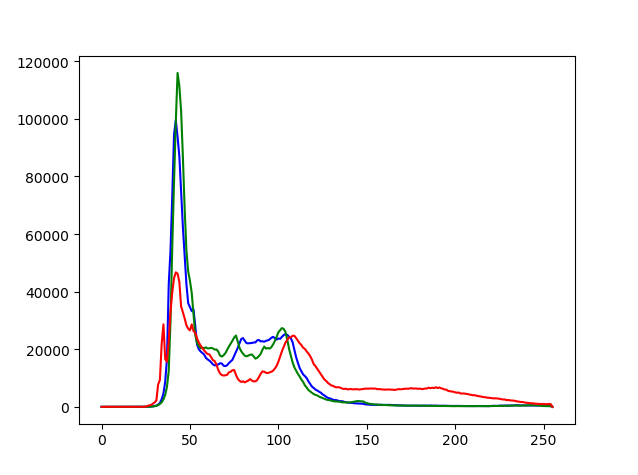
cv2.waitKey(0)

plt.plot(histb, 'b')

plt.plot(histg, 'g')

plt.plot(histr, 'r')

plt.show()

1. 三波段图的直方图
   * 1. 绘制子图

使用子图可以在一个窗口绘制多张图。

代码：

plt.subplot(2, 1, 1)

3个参数分别为：总行数，总列数，活跃区；绘图区域是按从左至右，从上至下的顺序编号，例如在4×4的方格上，活跃区6在方格上的坐标为(2, 2)。

代码：

#对第1子图进行设定

plt.subplot(3, 1, 1)

plt.plot(histb, 'b')

plt.xlabel('x')

plt.ylabel('y')

#对第2子图进行设定

plt.subplot(3, 1, 2)

plt.plot(histg, 'g')

plt.xlabel('x')

plt.ylabel('y')

#对第3子图进行设定

plt.subplot(3, 1, 3)

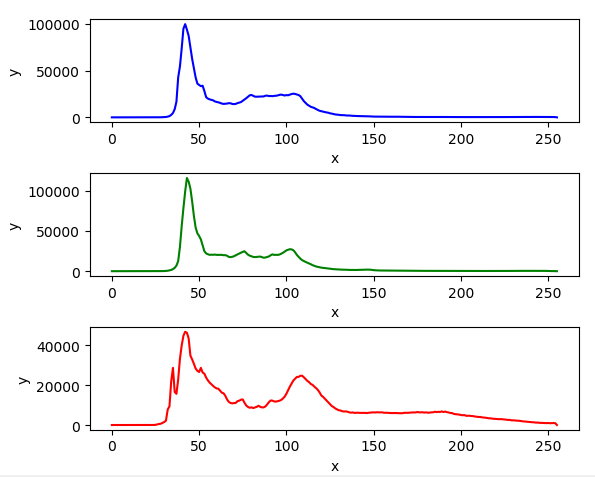
plt.plot(histr, 'r')

plt.xlabel('x')

plt.ylabel('y')

#显示图像

plt.show()



1. 子图为3行1列的直方图