Python 绘图以及文件的基本操作

https://pypi.org/

Python Package Index (PyPI)

Python 绘图库 Matplotlib

CMD 以管理员身份运行

```
pip install -U pip setuptools
pip install matplotlib
```

import numpy as np
from matplotlib import pyplot as plt

x = np.arange(1,11)
y = 2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y) plt.show()

https://matplotlib.org/

https://matplotlib.org/tutorials/index.html#introductory

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 2, 100)

plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')

plt.xlabel('x label')
plt.ylabel('y label')

plt.title("Simple Plot")

plt.legend()

plt.show()
```

https://matplotlib.org/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py

numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)

增加中文的支持：

```
#coding=utf-8
```

```
#coding=utf-8
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
import numpy as np
x = np.linspace(-2, 2, 200)

plt.plot(x, x, label='y=x')
plt.plot(x, x**2, label='y=x^2')
plt.plot(x, x**3, label='y=x^3')

plt.xlabel('x 轴')
plt.ylabel('y 轴')

plt.title("简单绘图")

plt.legend()

plt.show()
```

np.arange()
函数返回一个有终点和起点的固定步长的排列，如[1,2,3,4,5]，起点是 1，终点是 5，步长为 1。
参数个数情况： np.arange()函数分为一个参数，两个参数，三个参数三种情况
1）一个参数时，参数值为终点，起点取默认值 0，步长取默认值 1。
2）两个参数时，第一个参数为起点，第二个参数为终点，步长取默认值 1。
3）三个参数时，第一个参数为起点，第二个参数为终点，第三个参数为步长。其中步长支持小数

```
#coding=utf-8
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt1
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
import numpy as np
x = np.linspace(-2, 2, 200)
y=np.sin(x)
plt.plot(x, x, label='y=x')
plt.plot(x, x**2, label='y=x^2')
plt.plot(x, x**3, label='y=x^3')

plt.xlabel('x 轴')
plt.ylabel('y 轴')

plt.title("简单绘图")
```

```
plt.legend()
plt1.plot(x,y)

plt.show()
plt1.show()
```

python 可视化库 matplotlib 的显示模式默认为阻塞（block）模式
plt.show() 之后不会执行
plt.ion()这个函数，使 matplotlib 的显示模式转换为交互（interactive）模式
plt.show() 之后继续执行

plt.ioff()关闭交互模式
plt.ion()打开交互模式

```
import matplotlib.pyplot as plt
plt.ion()
plt.plot([1.6, 2.7])
plt.plot([3, 2])

# 创建一个画布
plt.figure()
# 在 figure 下线
plt.plot(x, y1, "-o") #实线
plt.plot(x, y2, "--o") #虚线
plt.plot(x, y3, "-.o") #虚点线
plt.plot(x, y4, ":o") # 点线
# 展现画布
plt.show()

plt.plot(x, y1, "-.") # 点
plt.plot(x, y2, "-,") # 像素点
plt.plot(x, y3, "-o") # 圆点
```

'^'　上三角点
'v'　下三角点
'<'　左三角点
'>'　右三角点

```
plt.plot(x, y1, "-^")
plt.plot(x, y2, "-v")
plt.plot(x, y3, "-<")
plt.plot(x, y4, "→")
```

'1'　下三叉点
'2'　上三叉点
'3'　左三叉点

'4' 右三叉点

```
plt.plot(x, y1, "-1")
plt.plot(x, y2, "-2")
plt.plot(x, y3, "-3")
plt.plot(x, y4, "-4")
```

's' 正方点
'p' 五角点
'*' 星形点
'h' 六边形 1
'H' 六边形 2

```
plt.plot(x, y1, "-s")
plt.plot(x, y2, "-p")
plt.plot(x, y3, "-*")
plt.plot(x, y4, "-h")
plt.plot(x, y5, "-H")
```

'+' 加号点
'x' 乘号点
'D' 实心菱形点
'd' 细菱形点
'_' 横线点
'|' 竖线点

```
plt.plot(x, y1, "-+")
plt.plot(x, y2, "-x")
plt.plot(x, y3, "-D")
plt.plot(x, y4, "-d")
plt.plot(x, y5, "-_")
```

color="green" 指定颜色为绿色

linestyle="dashed" 指定线形为 dashed 类型

marker="o" 指定标记类型为 o 点

markerfacecolor="blue"指定标记的颜色为蓝色

markersize=20 指定标记的大小为 20

```
plt.plot(x, y1, "-P")
plt.plot(x, y2, "-|")
plt.plot(x, y3, color="#000000")
plt.plot(x, y4, "-o", markersize=20)
plt.plot(x, y5, "-^", markerfacecolor="blue")
```

散点图
plt.scatter(x, y, s, c ,marker, alpha)

x,y: x 轴与 y 轴的数据

s: 点的面积

c: 点的颜色

marker: 点的形状

alpha: 透明度

随机数
x = np.random.randn(N)
y2 = x + np.random.randn(N)*0.5

1.plot(x, y, marker='D')表示绘制折线图，marker 设置样式菱形。
2.scatter(x, y, marker='s', color='r')绘制散点图，红色正方形。
3.bar(x, y, 0.5, color='c')绘制柱状图，间距为 0.5，原色。
4.hist(data,40,normed=1,histtype='bar',
facecolor='yellowgreen',alpha=0.75)直方图。
5.设置 x 轴和 y 轴的坐标值：
xlim(-2.5, 2.5) #设置 x 轴范围 ylim(-1, 1) #设置 y 轴范围
6.显示中文和负号代码如下：
plt.rcParams['font.sas-serig']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号

```
from matplotlib import pyplot as plt
import numpy as np
import math
x=np.linspace(-10,10,100)


fig = plt.figure()
ax1 = fig.add_subplot(231)
plt.plot(x,x)
plt.sca(ax1)
ax2 = fig.add_subplot(232)
plt.plot(x,x*x)
plt.sca(ax2)
ax3 = fig.add_subplot(233)
plt.plot(x,x**3)
plt.sca(ax3)
ax4 = fig.add_subplot(234)
plt.plot(x,np.sin(x))
```

```
plt.sca(ax4)
ax5 = fig.add_subplot(235)
plt.plot(x,np.cos(x))
plt.sca(ax5)
ax6 = fig.add_subplot(236)
plt.plot(x,x*x+x**3)
plt.sca(ax6)
plt.grid(True)
plt.show()
```

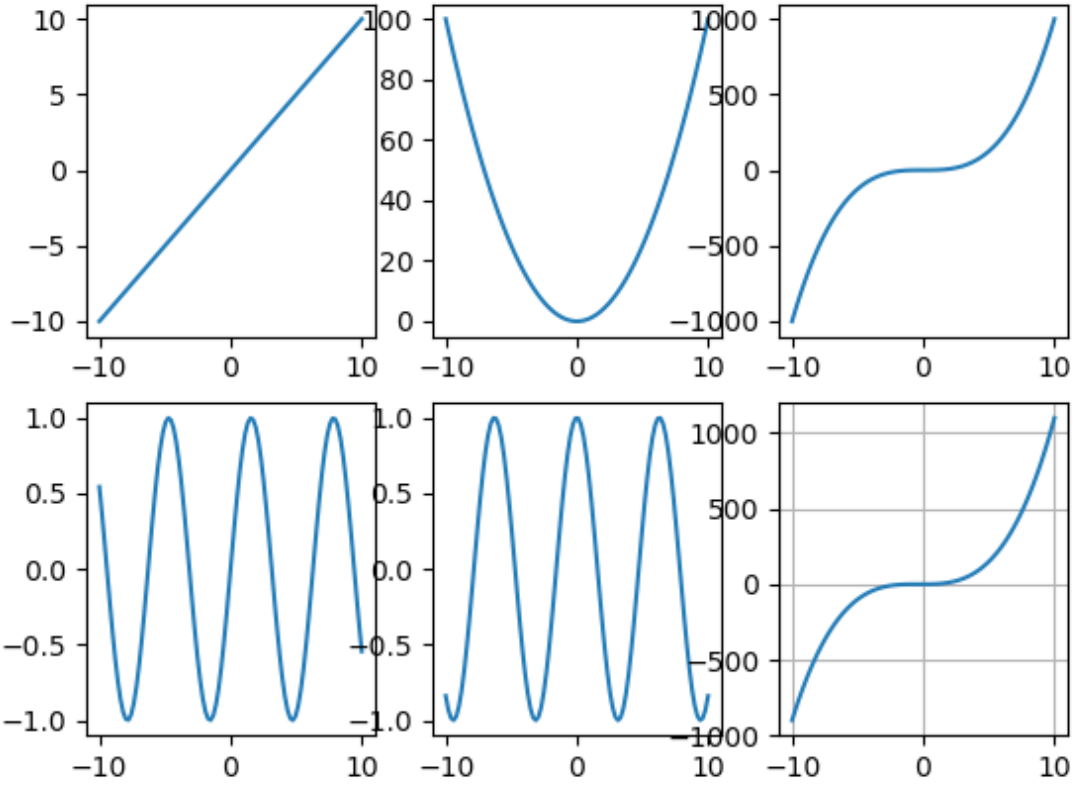
a.py - C:/Python/Python38/a.py (3.8.1)

File Edit Format Run Options Window Help

```
from matplotlib import pyplot as plt
import numpy as np
import math
x=np.linspace(-10,10,100)


fig = plt.figure()
ax1 = fig.add_subplot(231)
plt.plot(x,x)
plt.sca(ax1)
ax2 = fig.add_subplot(232)
plt.plot(x,x*x)
plt.sca(ax2)
ax3 = fig.add_subplot(233)
plt.plot(x,x**3)
plt.sca(ax3)
ax4 = fig.add_subplot(234)
plt.plot(x,np.sin(x))
plt.sca(ax4)
ax5 = fig.add_subplot(235)
plt.plot(x,np.cos(x))
plt.sca(ax5)
ax6 = fig.add_subplot(236)
plt.plot(x,x*x+x**3)
plt.sca(ax6)
plt.grid(True)
plt.show()
```

Figure 1



Python 的输入输出，文件目录小结

1，键盘输入

2，打印输出（显示器输出）

3，文件夹的建立

4，文件夹的命名

5，文件夹的删除

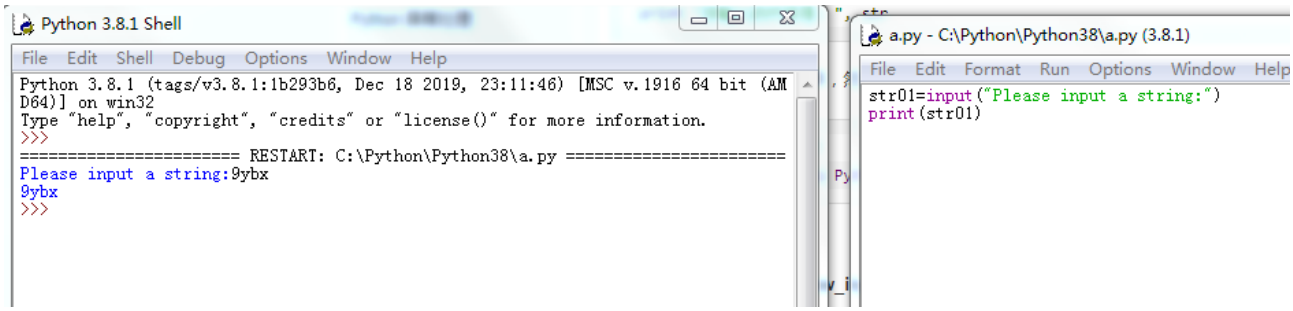
6，文件的删除

7，文件的输入

8，文件的输出

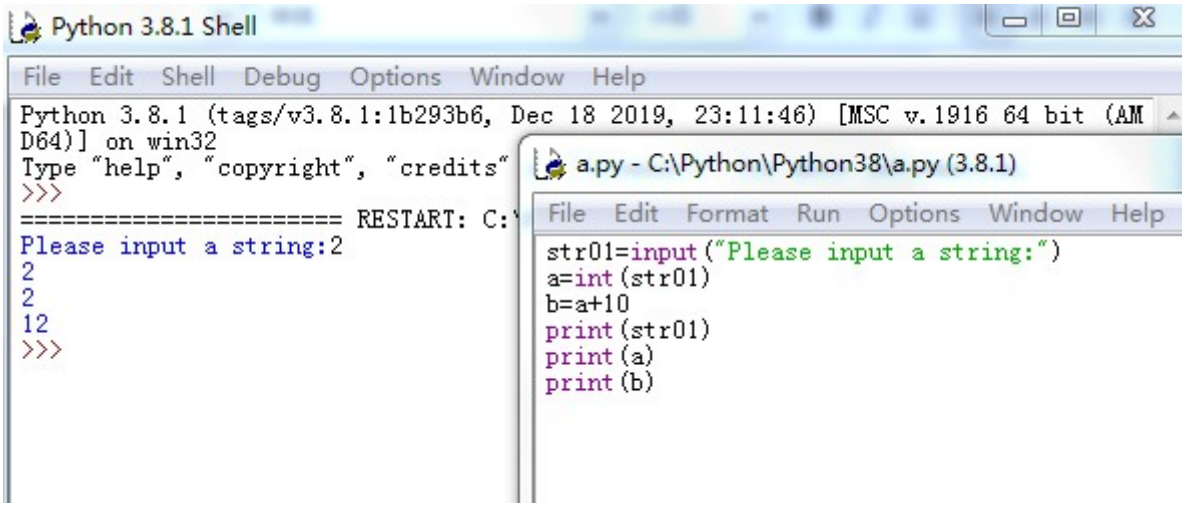
1，键盘输入

```
str01=input("Please input a string:")
print(str01)
```



将输入的字符串转换为整数

```
str01=input("Please input a string:")
a=int(str01)
b=a+10
print(str01)
print(a)
print(b)
```

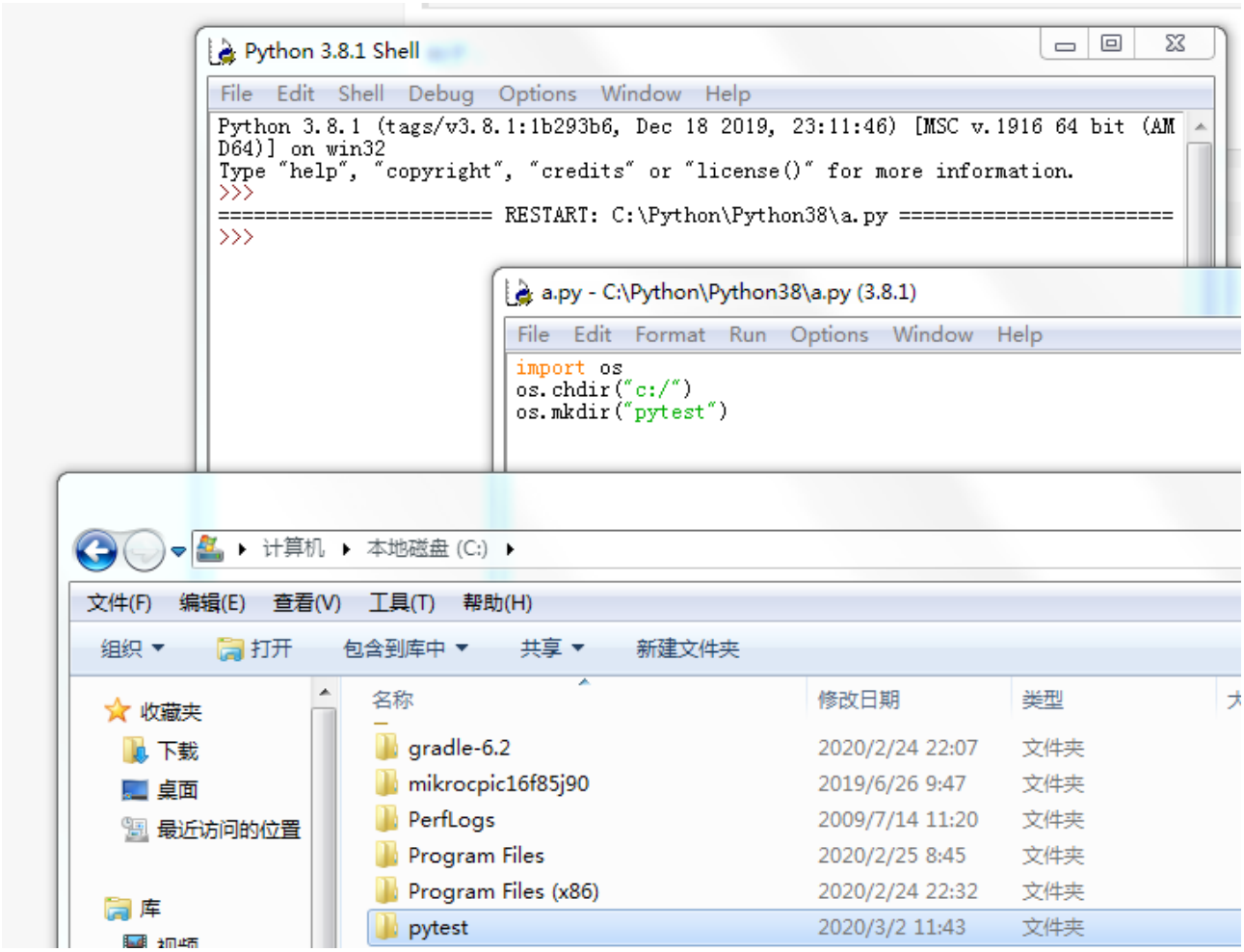


2，打印输出（显示器输出）

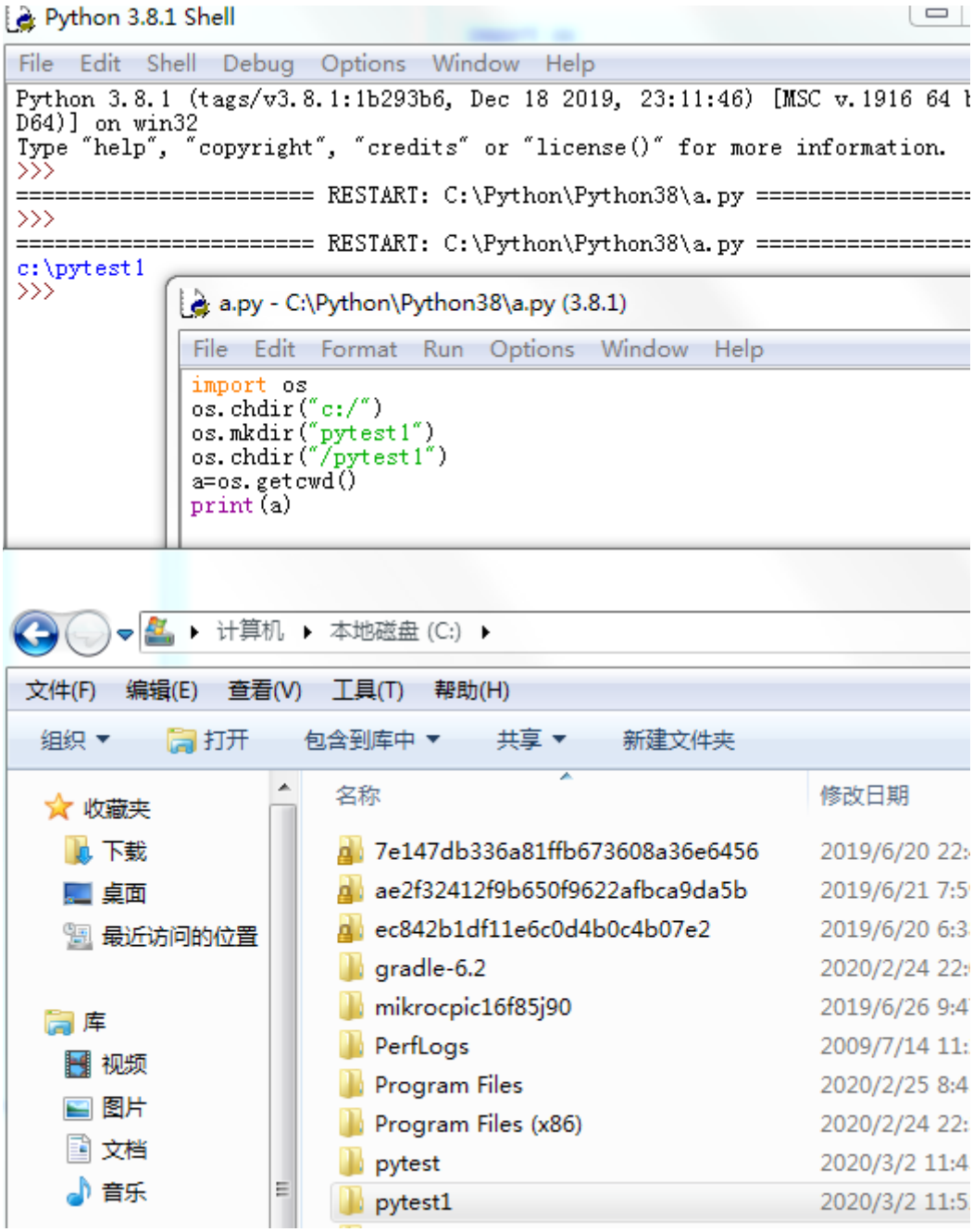
3，文件夹的建立

在 C 盘根目录下建立名为 pytest 的文件夹：

```
import os
os.chdir("c:/")
os.mkdir("pytest")
```




```
import os
os.chdir("c:/")
os.mkdir("pytest1")
os.chdir("/pytest1")
a=os.getcwd()
print(a)
```



4，文件夹的命名

```
os.rename( "test1.txt", "test2.txt" )
```

5，文件夹的删除

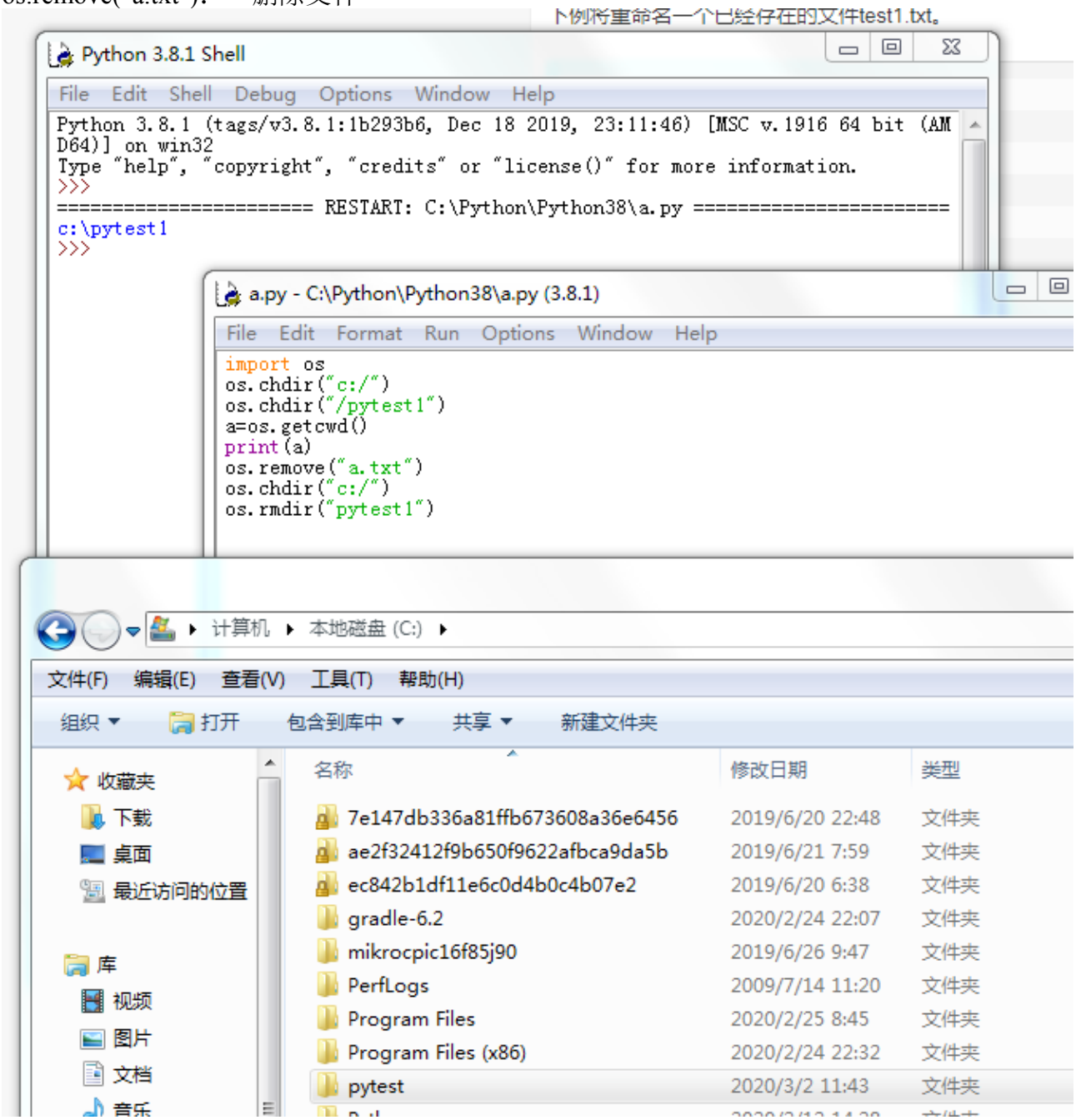
os.rmdir("pytest1")： 删除文件夹

6，文件的删除

```
import os
os.chdir("c:/")
os.chdir("/pytest1")
a=os.getcwd()
print(a)
os.remove("a.txt")
os.chdir("c:/")
```

os.rmdir("pytest1")

os.remove("a.txt")：　删除文件

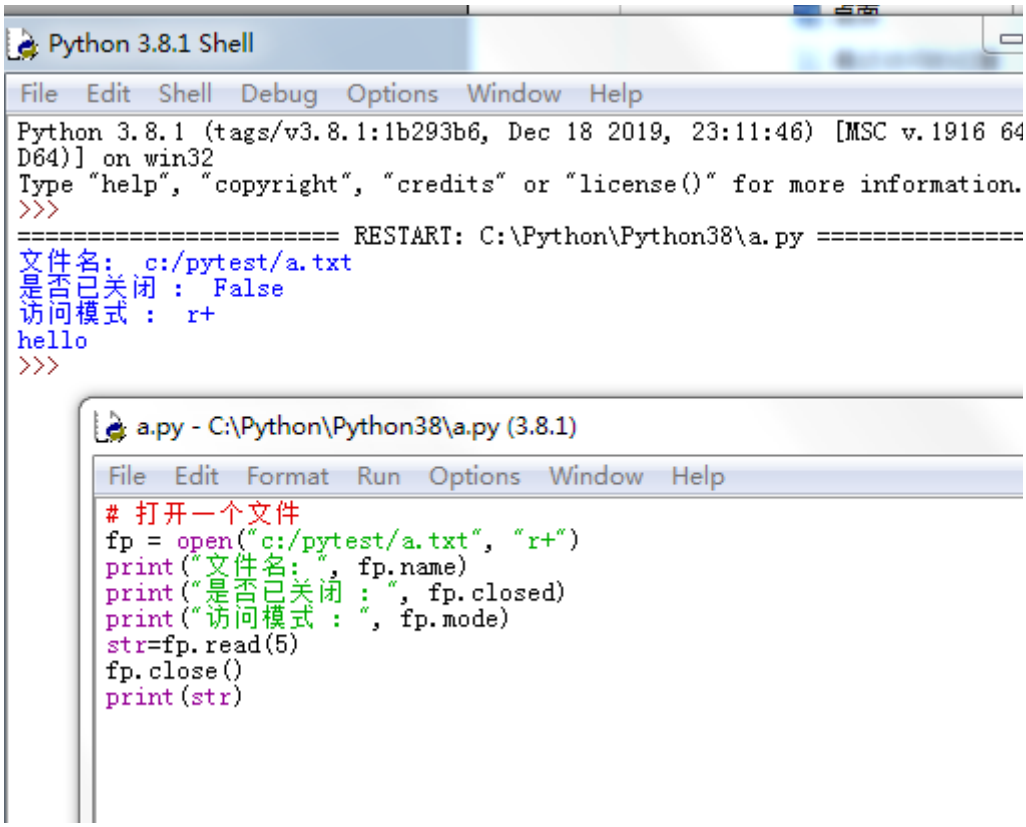


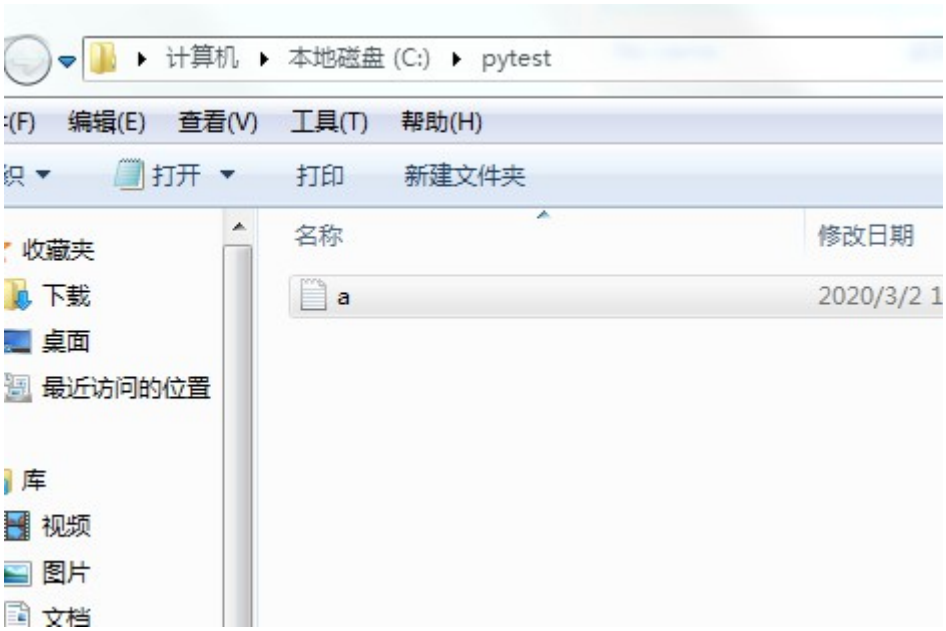
getcwd()方法显示当前的工作目录

7，文件的输入

```
# 打开一个文件
fp = open("c:/pytest/a.txt", "r+")
print("文件名: ", fp.name)
print("是否已关闭 : ", fp.closed)
print("访问模式 : ", fp.mode)
str=fp.read(5)
```

fp.close()
print(str)



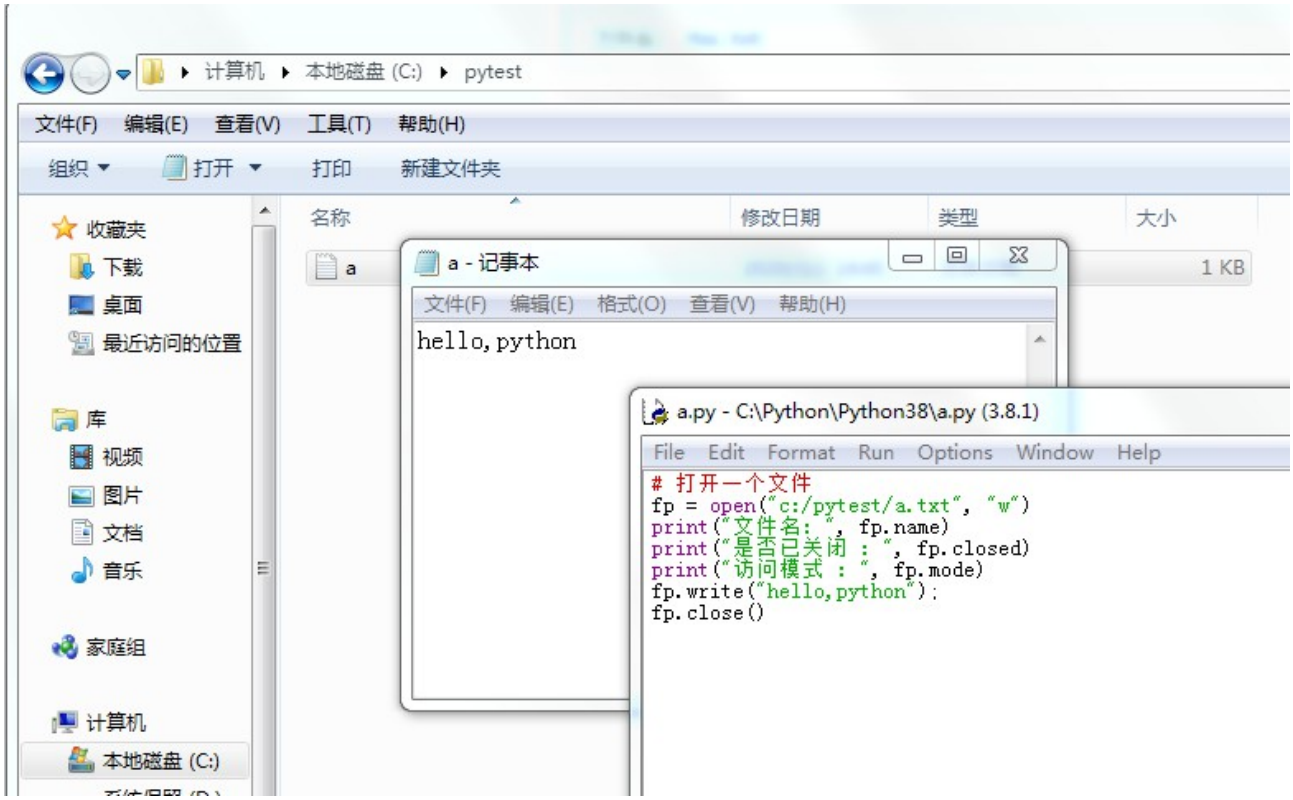


建立一个空的文件 a.txt 供程序调用。

8，文件的输出

```
# 打开一个文件
fp = open("c:/pytest/a.txt", "w")
print("文件名: ", fp.name)
```

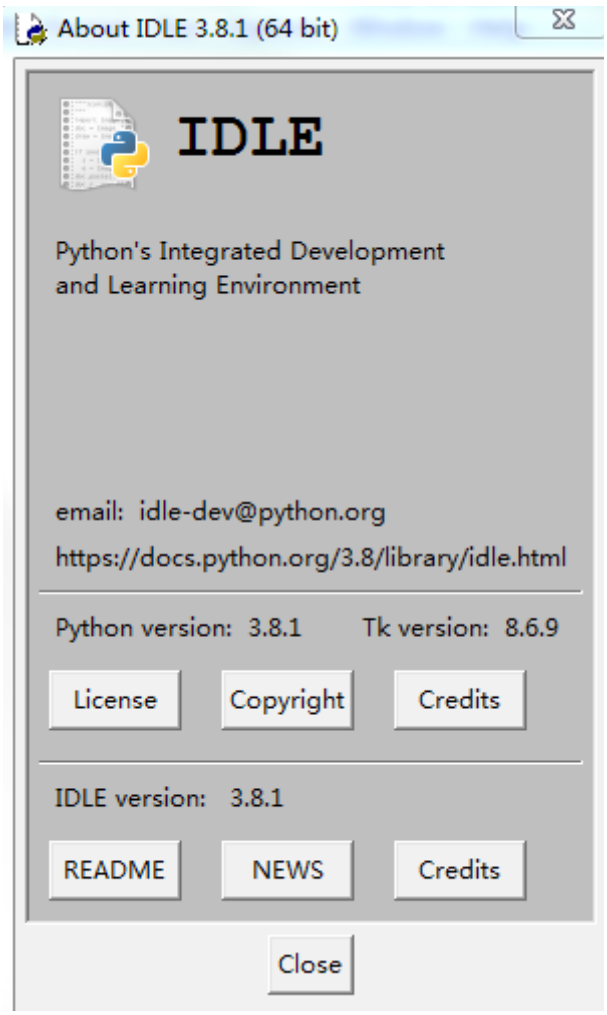
print("是否已关闭 : ", fp.closed)
print("访问模式 : ", fp.mode)
fp.write("hello,python");
fp.close()



OPENCV & python & 机器视觉

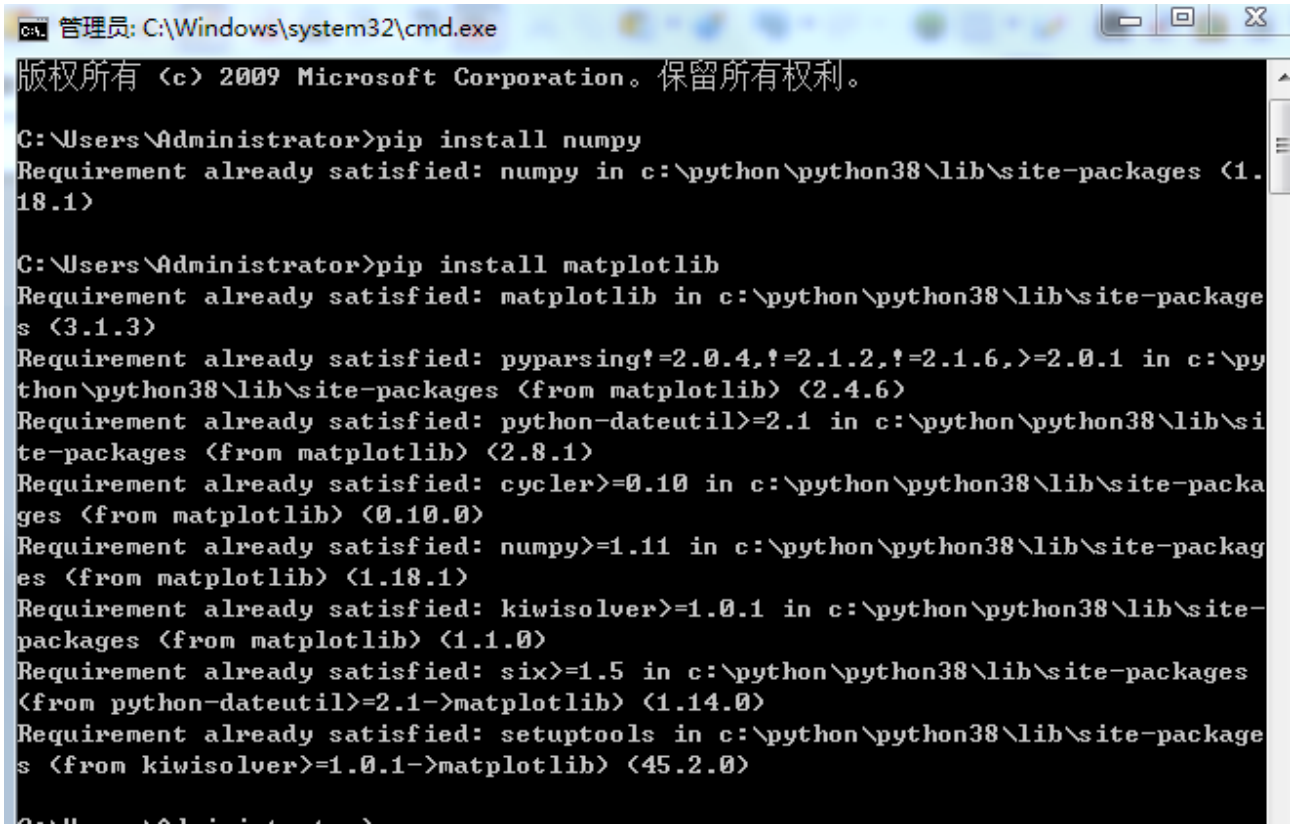
https://docs.opencv.org/4.2.0/d6/d00/tutorial_py_root.html

软件安装
首先安装 python



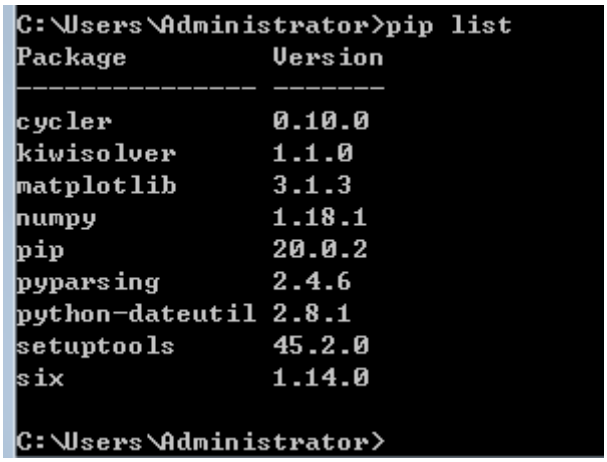
Numpy package (for example, using `pip install numpy` command).

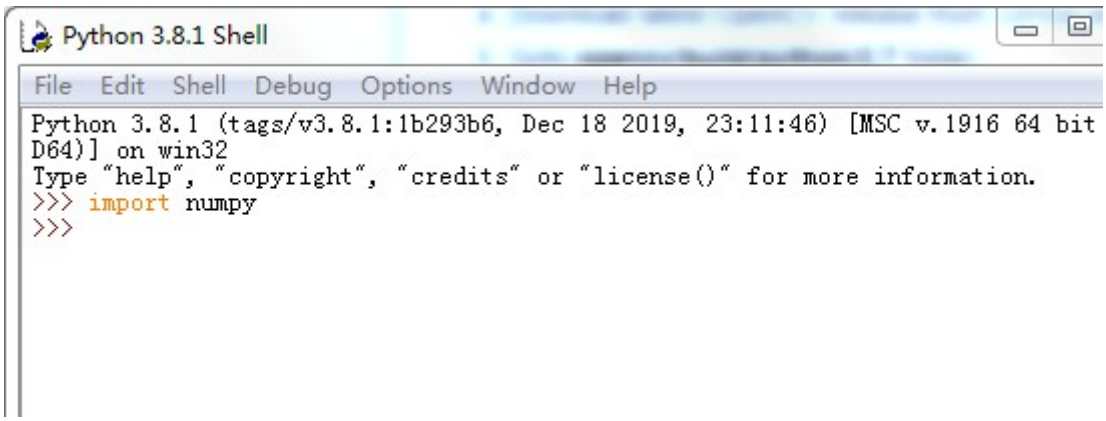
Matplotlib (`pip install matplotlib`) (Matplotlib is optional, but recommended since we use it a lot in our tutorials).



之前已经安装过了。

小知识：如何查找 python 环境中安装了什么？
命令：
pip list





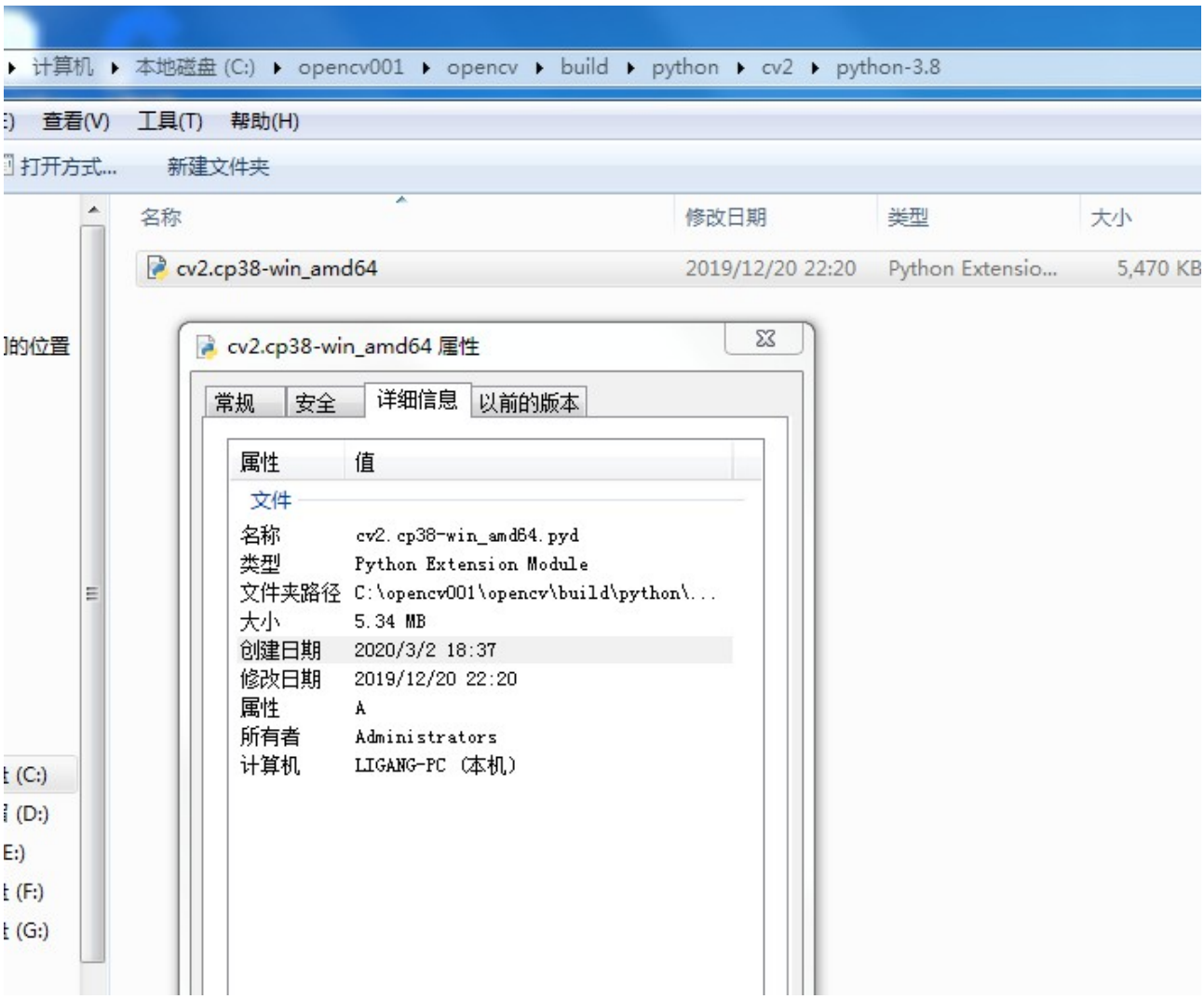
确认 numpy 是正确安装并可以使用。

https://sourceforge.net/projects/opencvlibrary/files/4.2.0/opencv-4.2.0-vc14_vc15.exe/download

下载并解压

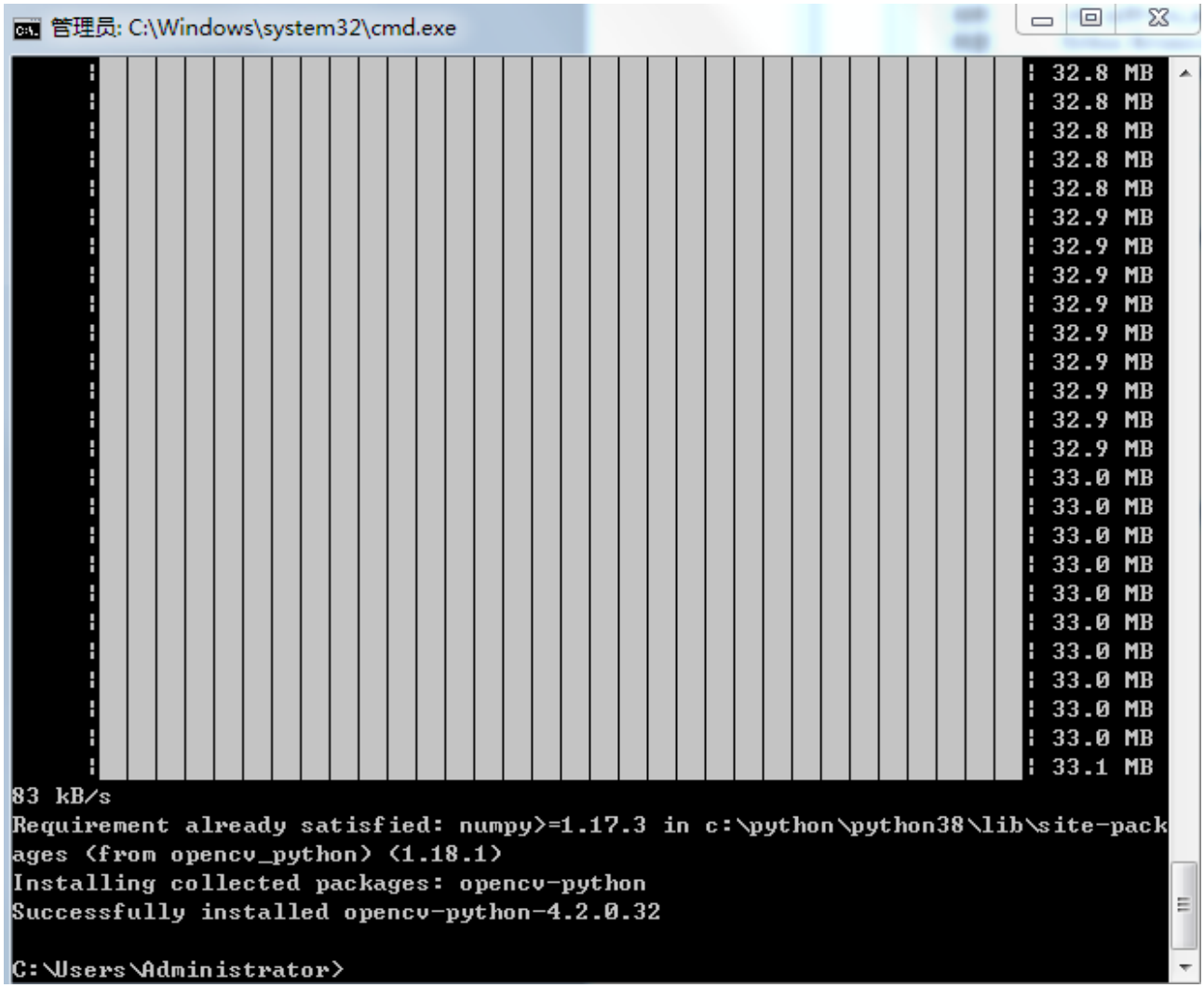
C:\opencv001\opencv\build\python\cv2\python-3.8

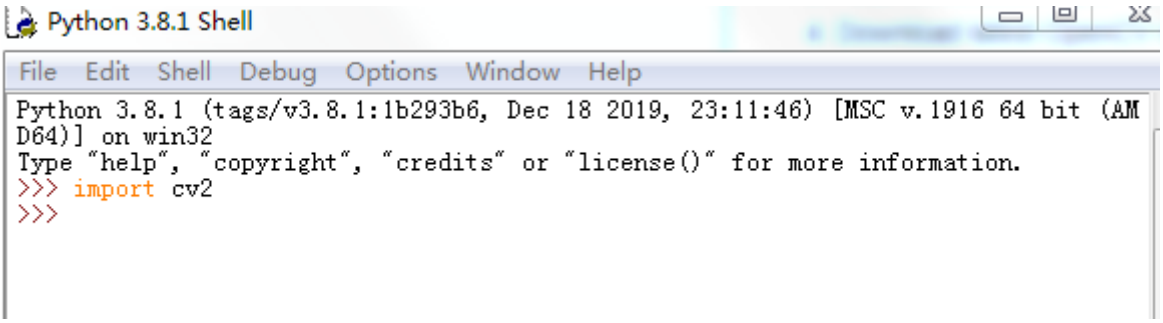
复制文件 cv2.pyd 到 C:\Python\Python38\Lib\site-packages 中



以上官方提供的方法可能会由于版本的不匹配不能安装，此时用下面的语句安装：

pip install opencv_python



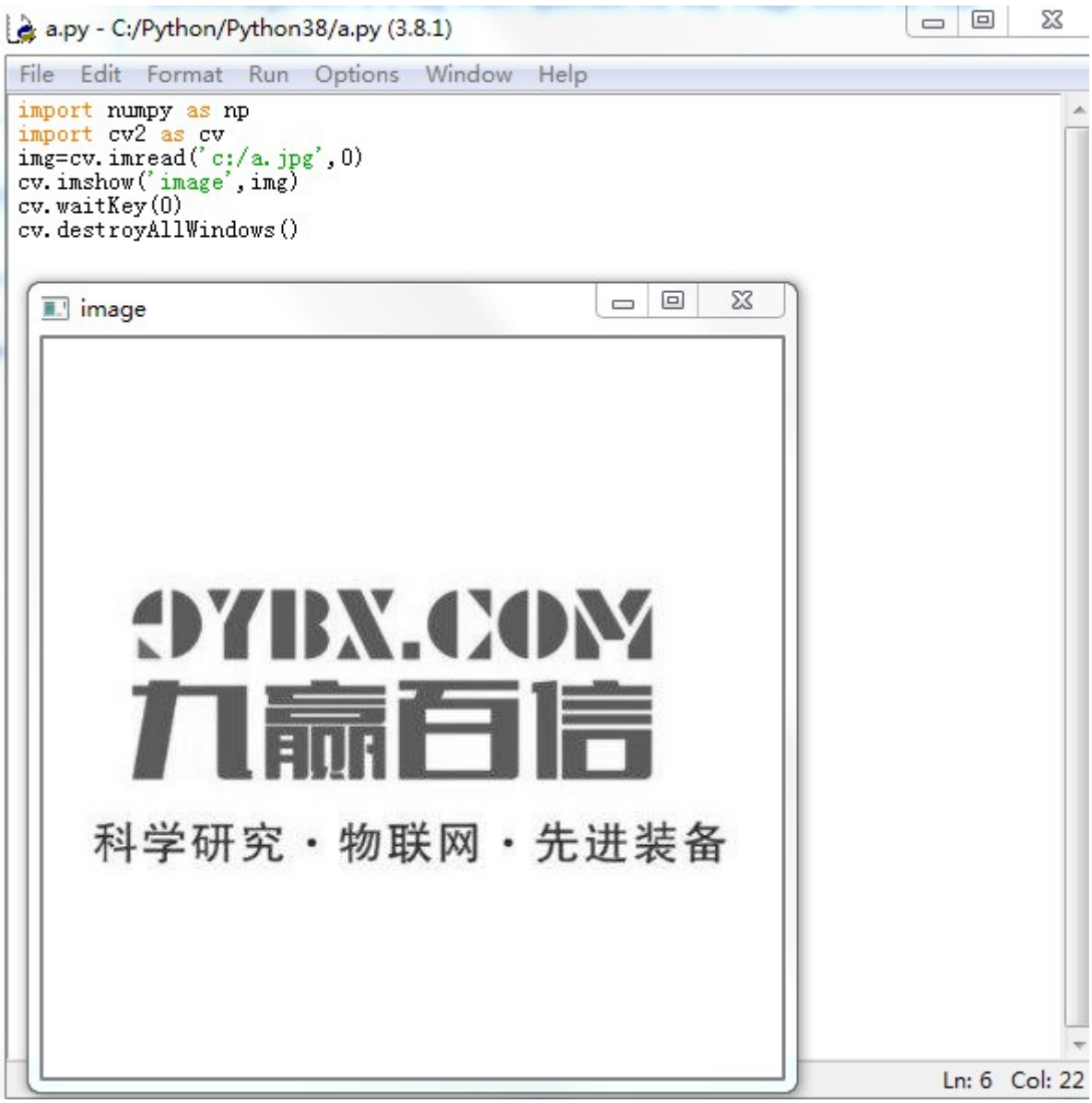


输入 import cv2 出现如上图所示，表明安装完毕。

测试功能：打开图像文件 c:/a.jpg

原图为彩色图像，打开为黑白图像。
代码如下：

```
import numpy as np
import cv2 as cv
img=cv.imread('c:/a.jpg',0)
cv.imshow('image',img)
cv.waitKey(0)
cv.destroyAllWindows()
```

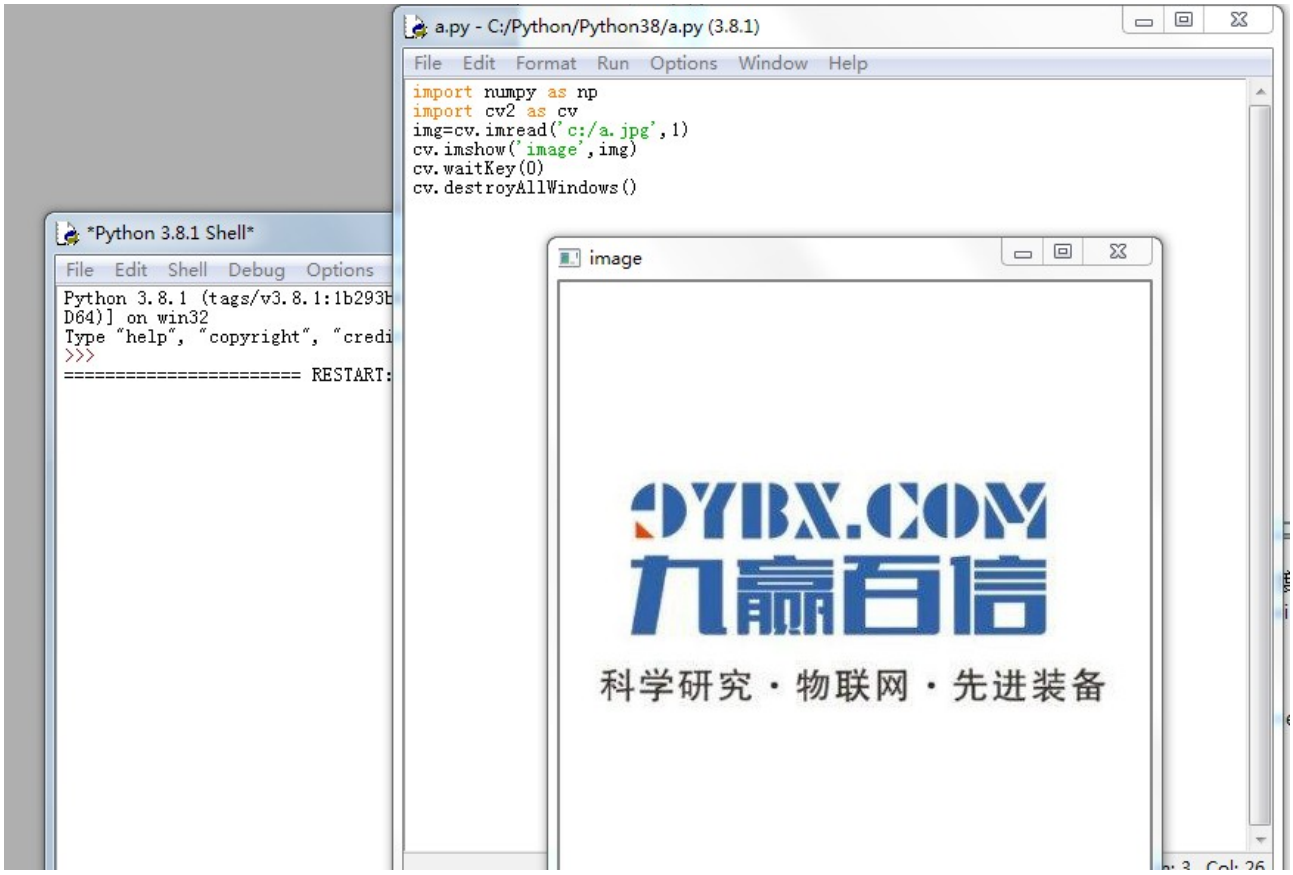


img=cv.imread('c:/a.jpg',0) 的参数含义：1：表示彩色, 0：表示灰度 or -1

cv.IMREAD_COLOR : Loads a color image. Any transparency of image will be neglected. It is the default flag.

cv.IMREAD_GRAYSCALE : Loads image in grayscale mode

cv.IMREAD_UNCHANGED : Loads image as such including alpha channel



彩色图像

将图像写入文件中

```
import numpy as np
import cv2 as cv
img=cv.imread('c:/a.jpg',0)
cv.imwrite('c:/a1.jpg',img)
cv.imshow('image',img)
cv.waitKey(0)
cv.destroyAllWindows()
```

原来彩色的图像，变成了灰度的图像了，这个可以作为图像转换的小应用。

使用 Matplotlib 库进行图像的显示，更多的显示方法，更多的参数选择，比如放大等

代码如下：

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img=cv.imread('c:/a.jpg',0)
plt.imshow(img,cmap='gray',interpolation='bicubic')
plt.xticks([]),plt.yticks([])
plt.show()
```

https://matplotlib.org/api/pyplot_api.html

https://blog.csdn.net/du_shuang/article/details/84111250

视频流的获取

```
import numpy as np
import cv2 as cv
cap = cv.VideoCapture(0)
if not cap.isOpened():
    print("Cannot open camera")
    exit()
while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    # Our operations on the frame come here
    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    # Display the resulting frame
    cv.imshow('frame', gray)
    if cv.waitKey(1) == ord('q'):
        break
# When everything done, release the capture
cap.release()
cv.destroyAllWindows()
```

a.py - C:/Python/Python38/a.py (3.8.1)

File Edit Format Run Options Window Help

```
import numpy as np
import cv2 as cv
cap = cv.VideoCapture(0)
if not cap.isOpened():
    print("Cannot open camera")
    exit()
while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    # Our operations on the frame come here
    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    # Display the resulting frame
    cv.imshow('frame', gray)
    if cv.waitKey(1) == ord('q'):
        break
# When everything done, release the capture
cap.release()
cv.destroyAllWindows()
```

CV_GRAY2RGB 是 gray 到 RGB
CV_BGR2GRAY,
CV_RGB2GRAY,
CV_GRAY2BGR,
CV_GRAY2RGB

色彩空间转换的模式，该 code 来实现不同类型的颜色空间转换。
比如 CV_BGR2GRAY 表示转换为灰度图，
CV_BGR2HSV 将图片从 RGB 空间转换为 HSV 空间。
其中当 code 选用 CV_BGR2GRAY 时，dst 需要是单通道图片。当 code 选用 CV_BGR2HSV 时，对于 8 位图，需要将 RGB 值归一化到 0-1 之间。这样得到 HSV 图中的 H 范围才是 0-360，S 和 V 的范围是 0-1

1、RGB 和 BGR（opencv 默认的彩色图像的颜色空间是 BGR）颜色空间的转换

cv::COLOR_BGR2RGB
cv::COLOR_RGB2BGR
cv::COLOR_RGBA2BGRA
cv::COLOR_BGRA2RGBA

2、向 RGB 和 BGR 图像中增添 alpha 通道

cv::COLOR_RGB2RGBA
cv::COLOR_BGR2BGRA

3、从 RGB 和 BGR 图像中去除 alpha 通道

cv::COLOR_RGBA2RGB
cv::COLOR_BGRA2BGR

4、从 RBG 和 BGR 颜色空间转换到灰度空间

cv::COLOR_RGB2GRAY
cv::COLOR_BGR2GRAY

cv::COLOR_RGBA2GRAY
cv::COLOR_BGRA2GRAY

5、从灰度空间转换到 RGB 和 BGR 颜色空间

cv::COLOR_GRAY2RGB
cv::COLOR_GRAY2BGR

cv::COLOR_GRAY2RGBA
cv::COLOR_GRAY2BGRA

6、RGB 和 BGR 颜色空间与 BGR565 颜色空间之间的转换

cv::COLOR_RGB2BGR565
cv::COLOR_BGR2BGR565
cv::COLOR_BGR5652RGB
cv::COLOR_BGR5652BGR
cv::COLOR_RGBA2BGR565
cv::COLOR_BGRA2BGR565
cv::COLOR_BGR5652RGBA
cv::COLOR_BGR5652BGRA

7、灰度空间域 BGR565 之间的转换

cv::COLOR_GRAY2BGR555
cv::COLOR_BGR5552GRAY

8、RGB 和 BGR 颜色空间与 CIE XYZ 之间的转换

cv::COLOR_RGB2XYZ
cv::COLOR_BGR2XYZ
cv::COLOR_XYZ2RGB
cv::COLOR_XYZ2BGR

9、RGB 和 BGR 颜色空间与 uma 色度（YCrCb 空间）之间的转换

cv::COLOR_RGB2YCrCb
cv::COLOR_BGR2YCrCb
cv::COLOR_YCrCb2RGB
cv::COLOR_YCrCb2BGR

10、RGB 和 BGR 颜色空间与 HSV 颜色空间之间的相互转换

cv::COLOR_RGB2HSV
cv::COLOR_BGR2HSV
cv::COLOR_HSV2RGB
cv::COLOR_HSV2BGR

11、RGB 和 BGR 颜色空间与 HLS 颜色空间之间的相互转换

cv::COLOR_RGB2HLS
cv::COLOR_BGR2HLS
cv::COLOR_HLS2RGB
cv::COLOR_HLS2BGR

12、RGB 和 BGR 颜色空间与 CIE Lab 颜色空间之间的相互转换

cv::COLOR_RGB2Lab
cv::COLOR_BGR2Lab
cv::COLOR_Lab2RGB
cv::COLOR_Lab2BGR

13、RGB 和 BGR 颜色空间与 CIE Luv 颜色空间之间的相互转换

cv::COLOR_RGB2Luv
cv::COLOR_BGR2Luv
cv::COLOR_Luv2RGB
cv::COLOR_Luv2BGR

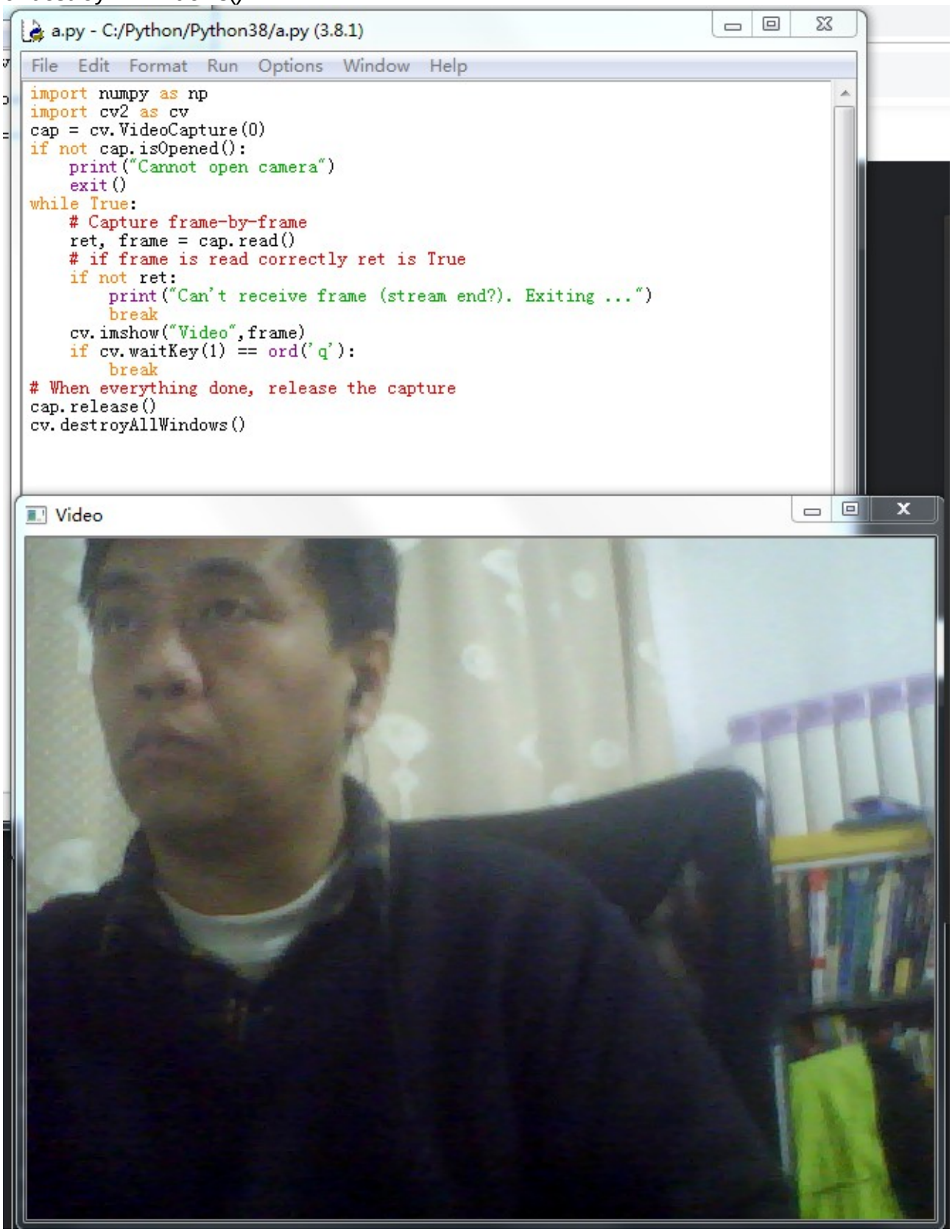
14、Bayer 格式（raw data）向 RGB 或 BGR 颜色空间的转换

cv::COLOR_BayerBG2RGB
cv::COLOR_BayerGB2RGB
cv::COLOR_BayerRG2RGB
cv::COLOR_BayerGR2RGB
cv::COLOR_BayerBG2BGR
cv::COLOR_BayerGB2BGR
cv::COLOR_BayerRG2BGR
cv::COLOR_BayerGR2BGR

彩色的视频捕捉

```
import numpy as np
import cv2 as cv
cap = cv.VideoCapture(0)
if not cap.isOpened():
    print("Cannot open camera")
    exit()
while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    cv.imshow("Video",frame)
    if cv.waitKey(1) == ord('q'):
        break
# When everything done, release the capture
cap.release()
```

cv.destroyAllWindows()



摄像头视频显示，以及捕捉录像

代码：

```
import numpy as np
import cv2 as cv
cap = cv.VideoCapture(0)
# Define the codec and create VideoWriter object
fourcc = cv.VideoWriter_fourcc(*'XVID')
out = cv.VideoWriter('c:/output.avi', fourcc, 20.0, (640,  480))
while cap.isOpened():
	ret, frame = cap.read()
	if not ret:
		print("Can't receive frame (stream end?). Exiting ...")
		break
    # frame = cv.flip(frame, 0)
	# write the flipped frame
	out.write(frame)
	cv.imshow('frame', frame)
	if cv.waitKey(1) == ord('q'):
		break
# Release everything if job is finished
cap.release()
out.release()
cv.destroyAllWindows()
```

https://docs.opencv.org/4.2.0/d6/d00/tutorial_py_root.html

https://docs.opencv.org/4.2.0/d1/dc5/tutorial_background_subtraction.html

a.py - C:/Python/Python38/a.py (3.8.1)

File Edit Format Run Options Window Help

```
import numpy as np
import cv2 as cv
cap = cv.VideoCapture(0)
# Define the codec and create VideoWriter object
fourcc = cv.VideoWriter_fourcc(*'XVID')
out = cv.VideoWriter('c:/output.avi', fourcc, 20.0, (640,  480))
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
   # frame = cv.flip(frame, 0)
    # write the flipped frame
    out.write(frame)
    cv.imshow('frame', frame)
    if cv.waitKey(1) == ord('q'):
        break
# Release everything if job is finished
cap.release()
out.release()
cv.destroyAllWindows()
```

生成的文件如下：

鼠标绘图测试代码

```
import numpy as np
import cv2 as cv
# mouse callback function
def draw_circle(event,x,y,flags,param):
    if event == cv.EVENT_LBUTTONDBLCLK:
        cv.circle(img,(x,y),100,(255,0,0),-1)
# Create a black image, a window and bind the function to window
img = np.zeros((512,512,3), np.uint8)
cv.namedWindow('image')
cv.setMouseCallback('image',draw_circle)
while(1):
    cv.imshow('image',img)
    if cv.waitKey(20) & 0xFF == 27:
```

```
            break
cv.destroyAllWindows()
```

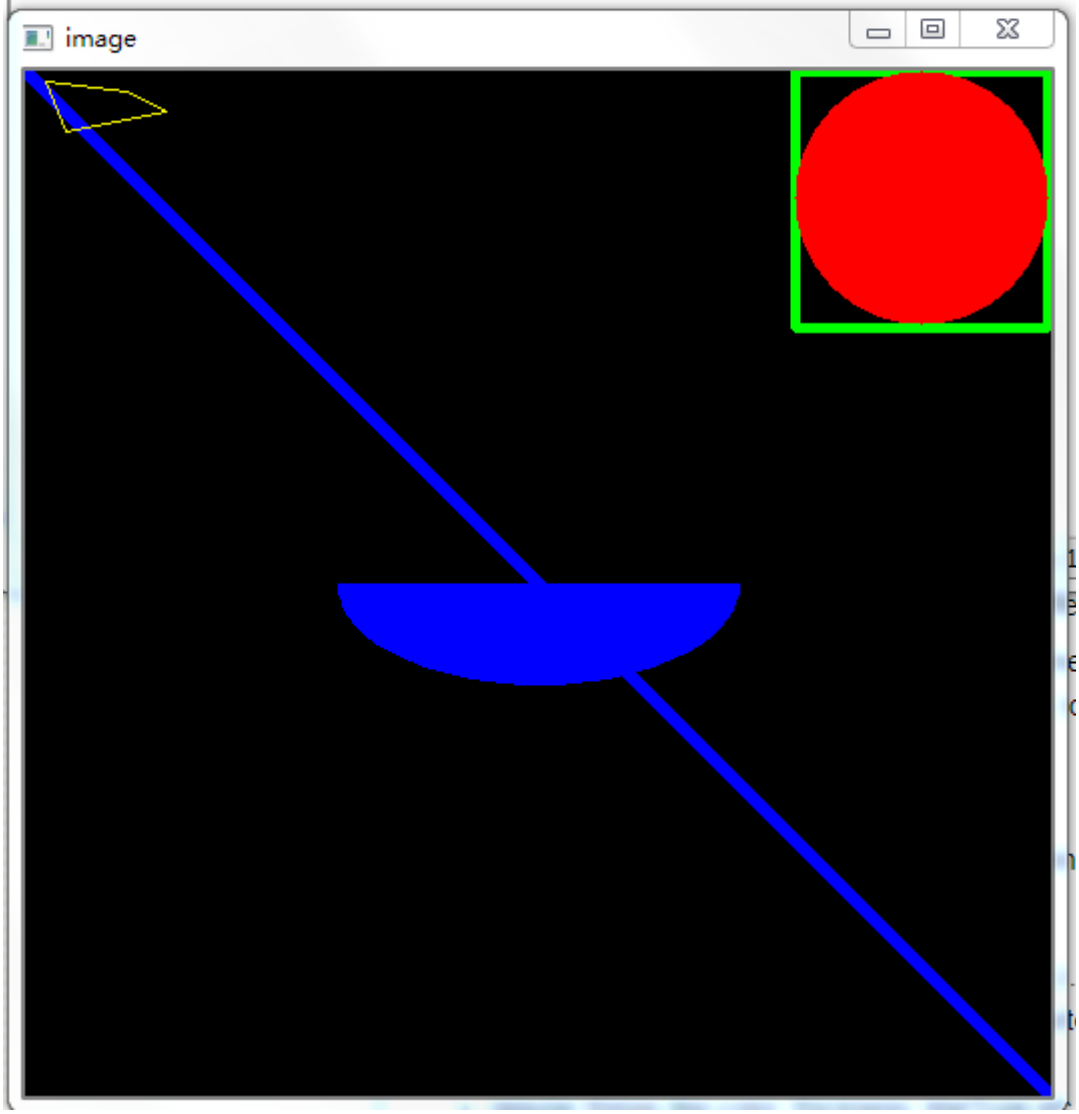
绘图：直线，圆，椭圆
https://docs.opencv.org/4.2.0/dc/da5/tutorial_py_drawing_functions.html
官网中少了后面的一些代码，添加后就 OK 了
代码如下：

```
import numpy as np
import cv2 as cv
# Create a black image
img = np.zeros((512,512,3), np.uint8)
# Draw a diagonal blue line with thickness of 5 px
cv.line(img,(0,0),(511,511),(255,0,0),5)
cv.rectangle(img,(384,0),(510,128),(0,255,0),3)
cv.circle(img,(447,63), 63, (0,0,255), -1)
cv.ellipse(img,(256,256),(100,50),0,0,180,255,-1)
pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)
pts = pts.reshape((-1,1,2))
cv.polylines(img,[pts],True,(0,255,255))
cv.namedWindow('image')
while(1):
    cv.imshow('image',img)
    if cv.waitKey(20) & 0xFF == 27:
        break
cv.destroyAllWindows()
```

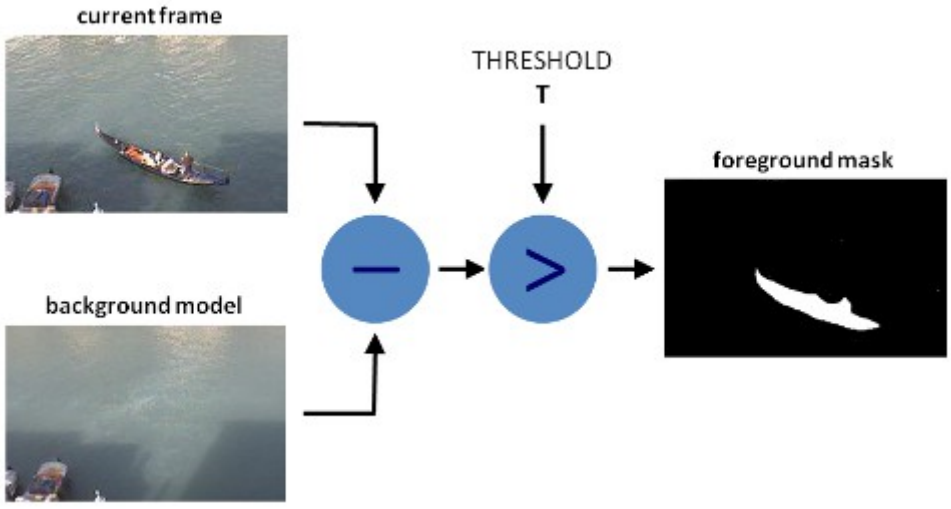
a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import numpy as np
import cv2 as cv
# Create a black image
img = np.zeros((512,512,3), np.uint8)
# Draw a diagonal blue line with thickness of 5 px
cv.line(img,(0,0),(511,511),(255,0,0),5)
cv.rectangle(img,(384,0),(510,128),(0,255,0),3)
cv.circle(img,(447,63), 63, (0,0,255), -1)
cv.ellipse(img,(256,256),(100,50),0,0,180,255,-1)
pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)
pts = pts.reshape((-1,1,2))
cv.polylines(img,[pts],True,(0,255,255))
cv.namedWindow('image')
while(1):
    cv.imshow('image',img)
    if cv.waitKey(20) & 0xFF == 27:
        break
cv.destroyAllWindows()
```



How to Use Background Subtraction Methods



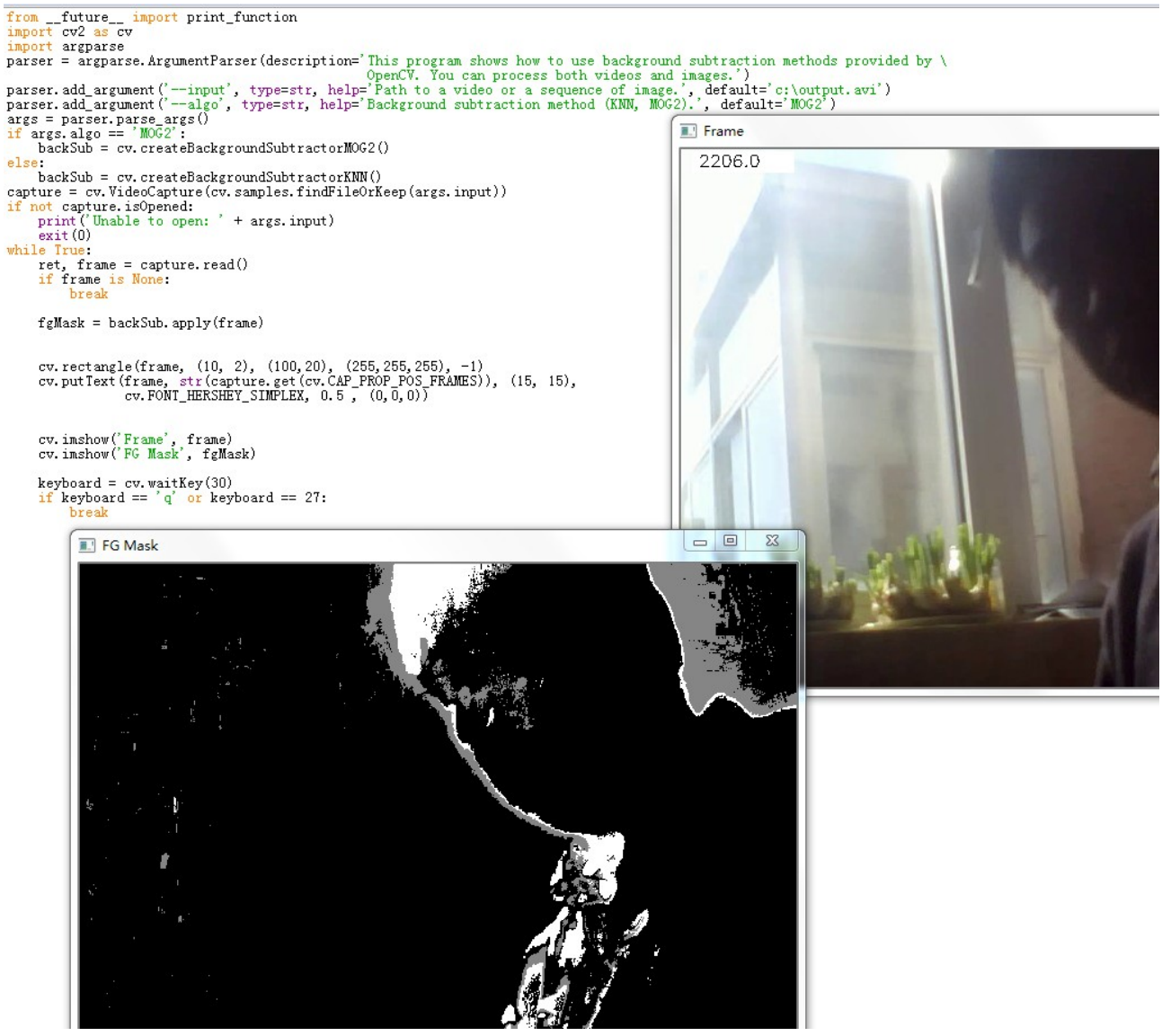

```
from __future__ import print_function
import cv2 as cv
import argparse
parser = argparse.ArgumentParser(description='This program shows how to use
background subtraction methods provided by \
                                                OpenCV. You can process both videos
and images.')
parser.add_argument('--input', type=str, help='Path to a video or a sequence of image.',
default='c:\output.avi')
parser.add_argument('--algo', type=str, help='Background subtraction method (KNN,
MOG2).', default='MOG2')
args = parser.parse_args()
if args.algo == 'MOG2':
    backSub = cv.createBackgroundSubtractorMOG2()
else:
    backSub = cv.createBackgroundSubtractorKNN()
capture = cv.VideoCapture(cv.samples.findFileOrKeep(args.input))
if not capture.isOpened:
    print('Unable to open: ' + args.input)
    exit(0)
while True:
    ret, frame = capture.read()
    if frame is None:
        break

    fgMask = backSub.apply(frame)
```

```
cv.rectangle(frame, (10, 2), (100,20), (255,255,255), -1)
cv.putText(frame, str(capture.get(cv.CAP_PROP_POS_FRAMES)), (15, 15),
            cv.FONT_HERSHEY_SIMPLEX, 0.5 , (0,0,0))


cv.imshow('Frame', frame)
cv.imshow('FG Mask', fgMask)

keyboard = cv.waitKey(30)
if keyboard == 'q' or keyboard == 27:
    break
```



图像中查找模式

在 a.jpg 中查找图像 a0.jpg

代码：

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
```

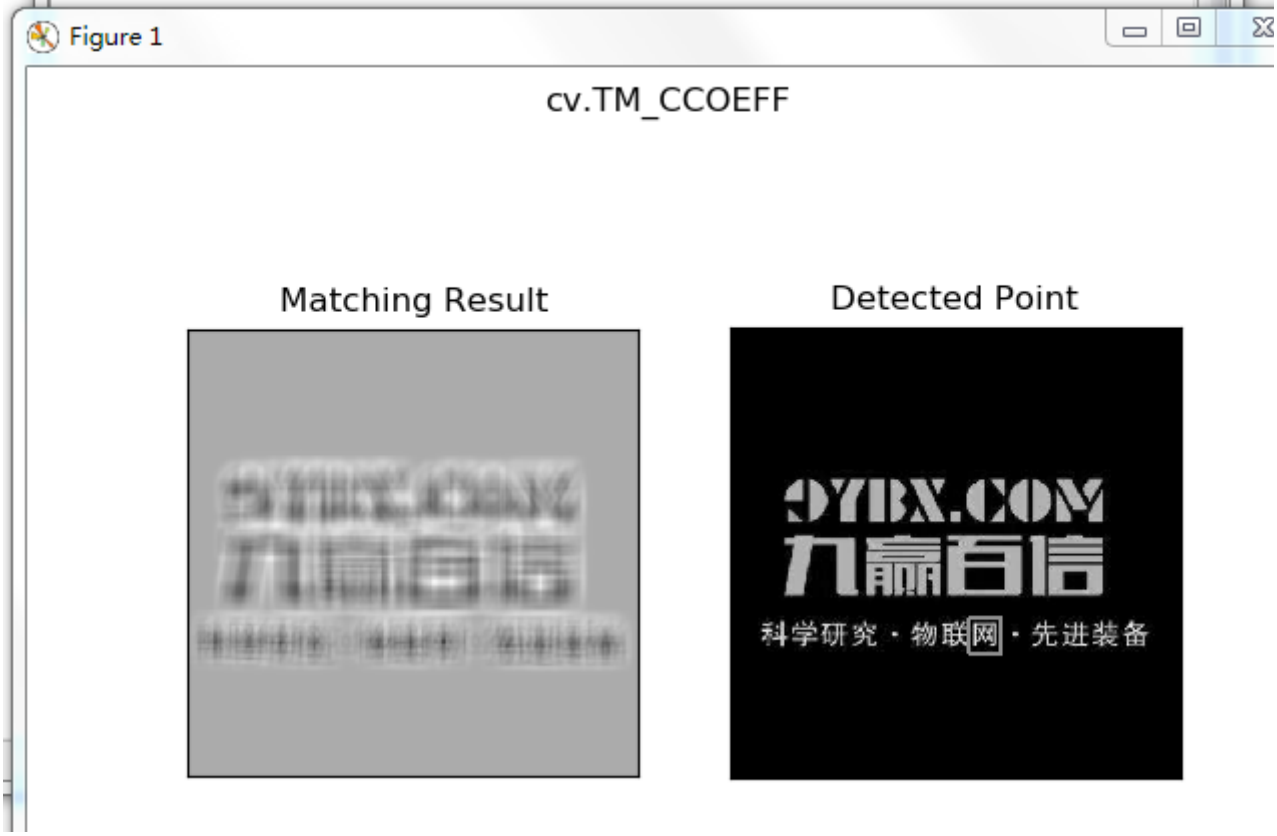
```
if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
    top_left = min_loc
else:
    top_left = max_loc
bottom_right = (top_left[0] + w, top_left[1] + h)
cv.rectangle(img,top_left, bottom_right, 122, 2)
plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
plt.suptitle(meth)
plt.show()
```

a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, 122, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
    plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
    plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
    plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
    plt.suptitle(meth)
    plt.show()
```

Figure 1



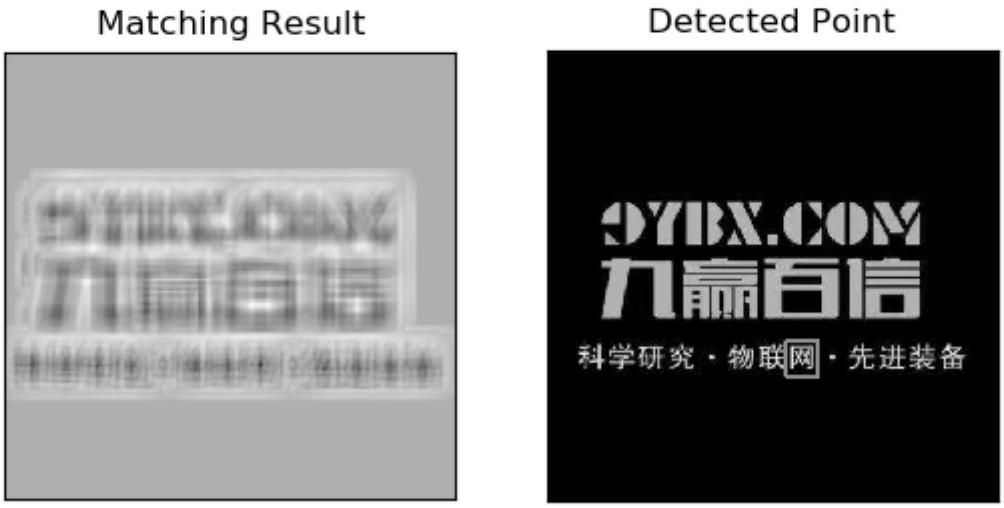
a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, 122, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
    plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
    plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
    plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
    plt.suptitle(meth)
    plt.show()
```

Figure 1

cv.TM_CCOEFF_NORMED



a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, 122, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
    plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
    plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
    plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
    plt.suptitle(meth)
    plt.show()
```

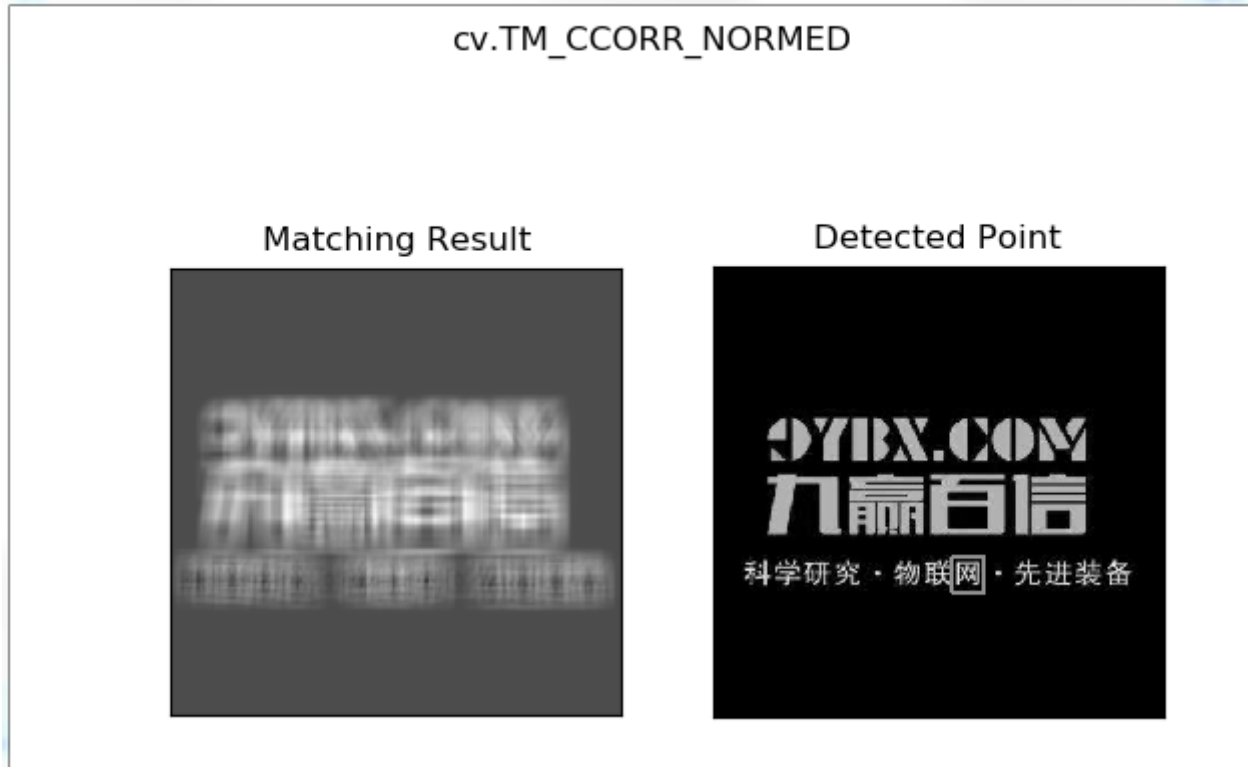
Figure 1

a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, 122, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
    plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
    plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
    plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
    plt.suptitle(meth)
    plt.show()
```

Figure 1



a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, 122, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
    plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
    plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
    plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
    plt.suptitle(meth)
    plt.show()
```

Figure 1

cv.TM_SQDIFF

a1.py - C:\Python\Python38\a1.py (3.8.1)

File Edit Format Run Options Window Help

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('c:/a.jpg',0)
img2 = img.copy()
template = cv.imread('c:/a0.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
            'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, 122, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray_r')
    plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
    plt.subplot(122),plt.imshow(img,cmap = 'gray_r')
    plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
    plt.suptitle(meth)
    plt.show()
```

Figure 1

cv.TM_SQDIFF_NORMED

methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']

六种模式的识别。

CV_TM_SQDIFF 平方差匹配法：该方法采用平方差来进行匹配；最好的匹配值为 0；匹配越差，匹配值越大。

CV_TM_CCORR 相关匹配法：该方法采用乘法操作；数值越大表明匹配程度越好。

CV_TM_CCOEFF 相关系数匹配法：1 表示完美的匹配；-1 表示最差的匹配。

CV_TM_SQDIFF_NORMED 归一化平方差匹配法

CV_TM_CCORR_NORMED 归一化相关匹配法

CV_TM_CCOEFF_NORMED 归一化相关系数匹配法

2.匹配算法

TM_SQDIFF	$$R(x,y)=\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2$$
TM_SQDIFF_NORMED	$$R(x,y)=\frac{\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$$
TM_CCORR	$$R(x,y)=\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))$$
TM_CCORR_NORMED	$$R(x,y)=\frac{\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$$
TM_CCOEFF	$$R(x,y)=\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))$$ where $$T'(x',y')=T(x',y')-1/(w\cdot h)\cdot\sum_{x'',y''}T(x'',y'')$$ $$I'(x+x',y+y')=I(x+x',y+y')-1/(w\cdot h)\cdot\sum_{x'',y''}I(x+x'',y+y'')$$
TM_CCOEFF_NORMED	$$R(x,y)=\frac{\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}T'(x',y')^2\cdot\sum_{x',y'}I'(x+x',y+y')^2}}$$

分别是：计算平方不同、计算归一化平方不同、计算相关性、计算归一化相关性、计算相关系数、计算归一化相关