

Lab 1的任务：

1. 安装并熟悉Jupyter Notebook或Jupyter Lab的环境（建议使用Anaconda来设置虚拟环境）
2. Python练习：熟悉计算机视觉中常用的Python语法
3. OpenCV中的图像读取、灰度化和二值化处理

Anaconda和Jupyter Notebook的安装

Jupyter Notebook或Jupyter Lab的安装请参看：<https://jupyter.org/install> (<https://jupyter.org/install>) 如果没有安装过Python，请先安装。Python 官网：<https://www.python.org/> (<https://www.python.org/>)；Python官方文档下载地址：<https://www.python.org/doc/> (<https://www.python.org/doc/>)。

建议安装Anaconda来管理Jupyter产品及相关的资源，并为这门课创建一个专门的虚拟环境(非必须，你可以用其他方法单独安装Python及Jupyter Notebook或Jupyter Lab，并安装相关的库)。

Anaconda支持Windows和MacOS系统。国内可以直接访问的镜像下载地址为：<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/> (<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>)。

Anaconda及虚拟环境的设置方法请参看：<https://blog.csdn.net/applebear1123/article/details/124544819> (<https://blog.csdn.net/applebear1123/article/details/124544819>) 或可在网上搜索其他教程。

请安装pandas、numpy、opencv-python、matplotlib、scipy、pandoc等常用库，可在Anaconda的图形界面里操作，也可以在notebook或终端里用pip install或conda install的命令进行操作（注意环境的设置）。安装完成后，打开Lab1文件夹中的**Lab1.ipynb**文件，完成下面的练习。

Python

本课程的后继任务大部分基于Python。本次联系的第一部分为Jupyter环境下的Python练习。不熟悉Python的同学，请跟随[W3School的Python教程](https://www.w3ccoo.com/python/default.html) (<https://www.w3ccoo.com/python/default.html>)练习Python的基本语法，掌握缩进规则。需掌握常见的数据类型和结构（tuple、array、dict、list等）、运算符、循环语句、判断语句、数据读写、函数、字符串操作、随机数生成、正态分布生成、Numpy、Scipy、Matplotlib绘图的基本方法。

Python部分的作业

第1题

完成名为“is_prime”的函数（替换掉“pass”），传入参数类型为int。如果传入的参数是质数，则返回True，否则返回False。然后编写一个循环，并利用该函数，打印1到1000之间的所有质数。

第2题

完成名为“list_process”的函数（替换掉“pass”）。该函数应传入一个由字符串组成的list，要求逐次打印（print）其中的每一个元素（字符串），且如果某个元素是数字，则该元素要连续打印三次；如果是其他情况，则该元素只打印一次，并且在其结尾处添加“#”。

例如，当输入的list为["1","-7","AEE0","8","Test"]，则应依次打印出：

111

-7-7-7

AEE0#

888

Test#

第3题

完成名为“greatest_common_divisor”的函数（替换掉“pass”），它传入两个整数，并计算它们的最大公约数。

第4题

完成名为“fibonacci”的函数（替换掉“pass”），它传入一个整数n，并从1开始，依次打印出[斐波纳契数列](https://wiki.mbalib.com/wiki/%E8%8F%B2%E6%B3%A2%E7%BA%B3%E5%A5%91%E6%95%B0%E5%88%97) (<https://wiki.mbalib.com/wiki/%E8%8F%B2%E6%B3%A2%E7%BA%B3%E5%A5%91%E6%95%B0%E5%88%97>)的前n个数。例如当n为6时，依次打印出1,1,2,3,5,8。

第5题

完成名为“mod_matrix”的函数（替换掉“pass”），它输入两个矩阵X和Y。

如X和Y的维度相同，则进行以下操作：

1. 计算X与Y的点乘结果，并将其转置，得到矩阵A。
2. 计算X与Y的叉乘结果，得到矩阵B。
3. 将A和B相加，得到矩阵C。输出C中的众数。

如果输入X和Y的维度不同，则输出0。

```
In [ ]: # 第1题
# 把pass替换为你自己的代码，完成is_prime函数:
def is_prime(number):
    pass

# 在下方编写用于打印1到1000之间所有质数的代码:
```

```
In [ ]: # 第2题
# 把pass替换为你自己的代码，完成list_process函数:
def list_process(mylist):
    pass

# 完成后，运行下面的代码
test = ["3", "18", "1KiP", "-1", "Top", "-E", "0"]
list_process(test)
```

```
In [ ]: # 第3题
# 把pass替换为你自己的代码，完成greatest_common_divisor函数:
def greatest_common_divisor(num1, num2):
    pass

# 完成后，运行下面的代码
x = 4840
y = 1600
print(x, "和", y, "的最大公约数是: ")
print(greatest_common_divisor(x, y))
```

```
In [ ]: # 第4题
# 把pass替换为你自己的代码，完成fibonacci函数:
def fibonacci(n):
    pass

# 完成后，运行下面的代码
fibonacci(13)
```

```
In [ ]: import numpy as np

# 第5题
# 把pass替换为你自己的代码，完成mod_matrix函数:
def mod_matrix(X, Y):
    pass

# 完成后，运行下面的代码
X = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Y = [[9, 8, 7], [6, 5, 4], [3, 2, 1]]
print(mod_matrix(X, Y))
```

图像部分：图片的读取、灰度化、二值化处理

导入图片

```
In [ ]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
In [ ]: # 用OpenCV的功能读取图片
img = cv2.imread("lab1_img/cat.jpg", cv2.IMREAD_COLOR)
```

```
In [ ]: # 查看图像的长宽维度（像素）
img.shape
```

```
In [ ]: # 在Notebook内部展示图片，可以使用plt.imshow()功能
plt.imshow(img)
```

```
In [ ]: # 看起来图片的颜色很奇怪
# 这是因为，cv2读取图片后会以BGR的顺序排列色彩通道，但plt的读取顺序是RGB
# 因此，需要转换三个色彩通道的顺序后再绘图
RGB_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(RGB_img)
```

```
In [ ]: # 将原本的彩色图片转换为灰度
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# 查看灰度图片的维度（注意它的通道数）
gray.shape
```

```
In [ ]: # 预览图片
plt.imshow(gray)
```

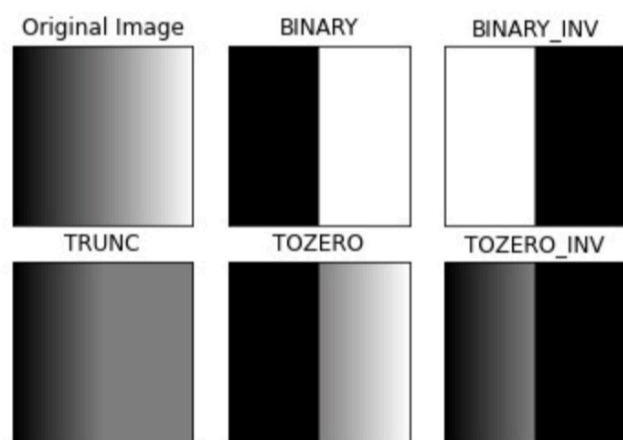
```
In [ ]: # 看起来这张图并不是灰度，但实际上这仍然是CV2的默认通道问题导致的，再次进行通道调整就能预览灰度图。
# 但需要注意的是，调整通道顺序的图片只用于预览，因为它改变了通道深度。后续的cv2图像相关操作，需要用gray进行。
gray_cvt = cv2.cvtColor(gray, cv2.COLOR_BGR2RGB)
plt.imshow(gray_cvt)
```

```
In [ ]: # 现在，对图片进行二值化处理，并用plt预览
# cv2.threshold传入参数分别为：图片（灰度）、阈值、超过阈值像素的设置值、阈值类型
ret, bw_img = cv2.threshold(gray, 100, 255, cv2.THRESH_BINARY)
plt.imshow(bw_img)
```

```
In [ ]: # 二值化的图片同样需要进行通道转换才能正确显示
bw_cvt = cv2.cvtColor(bw_img, cv2.COLOR_BGR2RGB)
plt.imshow(bw_cvt)
```

阈值的设置

在OpenCV中，Threshold(二值化)设置的种类如下：



(CV2.THRESH_BINARY_INV)

- 截断阈值化处理 (CV2.THRESH_TRUNC)
- 超阈值零处理 (CV2.THRESH_TOZERO_INV)
- 低阈值零处理 (CV2.THRESH_TOZERO)

思考

这些阈值设置分别应该用于什么情景下？

尝试用其他几种方法处理这张猫的图片

图像部分的作业

第1题

对lab1_img文件夹中的cat2.jpg进行灰度化和二值化操作。导入图片后，先预览并观察原图片的背景特点，然后自行选择二值化的设置种类，并调节阈值的设定值，让预览图中形成“白底+猫的黑色剪影”效果。请在下面的第一个Cell中录入及预览图片，在第二个Cell中处理图片，并预览处理后的图片。要求：白底+猫的剪影的效果要清晰、明显。

```
In [ ]: # 读取及预览图片
```

Binary	$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$	• 二值化阈值处理 (CV2.THRESH_BINARY)
Inverted Binary	$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$	• 反二值化阈值处理
Truncated	$\text{dst}(x, y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$	
To Zero	$\text{dst}(x, y) = \begin{cases} \text{src}(x, y) & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$	
To Zero Inverted	$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$	

```
In [ ]: # 对图片进行二值化处理, 并预览处理后的图片
```

第2题

对lab1_img文件夹中的city.jpg城市照片进行灰度化和二值化操作。导入图片后, 先预览并观察原图片的背景特点, 然后自行选择二值化的设置种类, 并调节阈值的设定值, 让预览图中形成“白底+城市建筑的黑色剪影”效果。请在下面的第一个Cell中录入及预览图片, 在第二个Cell中处理图片, 并预览处理后的图片。要求: 白底+黑色城市建筑剪影的效果要清晰、明显。

```
In [ ]: # 读取及预览图片
```

```
In [ ]: # 对图片进行二值化处理, 并预览处理后的图片
```

提交方式

完全所有Cell的运行后, 保存为ipynb和PDF格式(保留所有输出)。将导出的ipynb命名为“Lab1+姓名+学号.ipynb”, 将导出的PDF命名为“Lab1+姓名+学号.pdf”, 并将上述两个文件提交到学习通作业模块的相应位置。PDF格式文件可以从页面左上角的“File/文件”菜单里选择通过HTML或LaTeX导出, 也可以通过打印预览, 保存为PDF。请独立完成练习, 参考答案将在截止时间后公布。截止时间: 2024年5月10日23:59