

# 文华班《数字图像处理》实验一附录

## 1、Python 环境搭建与第一个 Python 程序

Python 是一种流行的解释性编程语言,它具有语法简单、优雅的特点。同时由于 Python 是跨平台的,它可以运行在 Windows、Mac 和各种 Linux/Unix 系统上。

首先第一步,我们要先学会如何在自己的电脑上搭建 Python 环境。我们提供两种在 Windows 系统下 Python 安装方式,分别是直接安装与 Anaconda 安装,其余系统如 Mac 等安装方式类似。

**1) 直接安装:** 首先从 Python 官网 (<https://www.python.org/downloads/>) 下载对应版本的 Python 安装程序,此处我们推荐下载 Python3.8 的 64 位的安装程序。我们选择下载 Windows x86-64 executable installer。详细的安装过程可以参考图 1-1-1 至 1-1-2。



图 1-1-1 Python 安装界面, 要勾选 Add Python 3.8 to PATH



图 1-1-2 Python 安装完成界面, 点击 Close 即可

**2) Anaconda 安装:** 首先从 Anaconda 官网 (<https://www.anaconda.com/products/individual>) 下载对应系统所需的安装包,例如 Windows 用户下载对应的 Python3.8 的 64 位的图形安装程序 (64-Bit Graphical Installer)。详细的安装过程可以参考图 1-2-1 至 1-2-8。

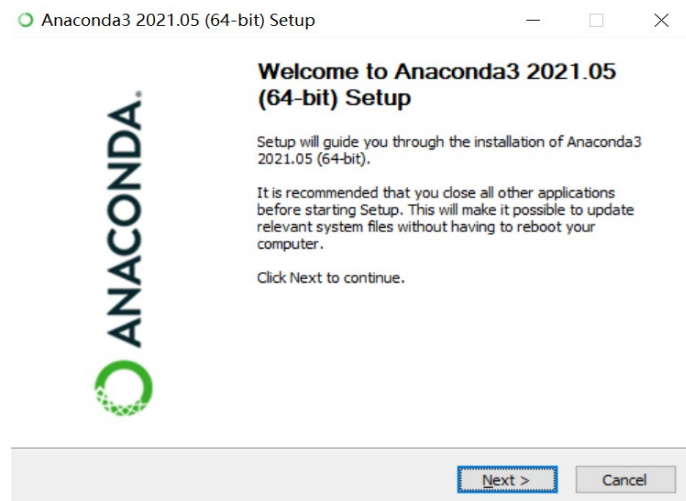


图 1-2-1 Anaconda 安装界面

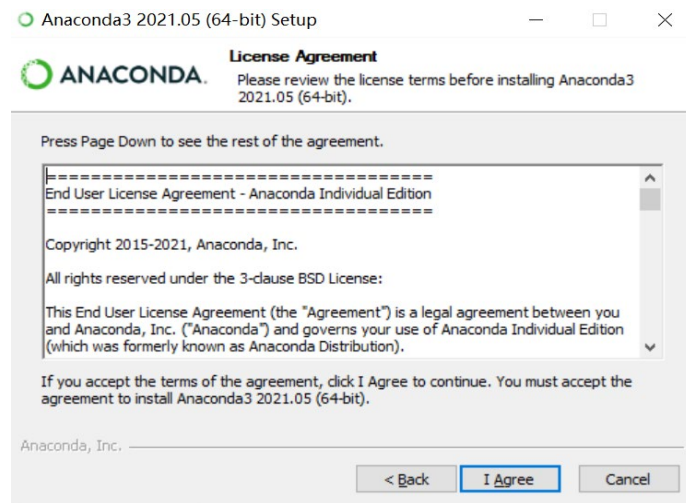


图 1-2-2 Anaconda 许可协议界面

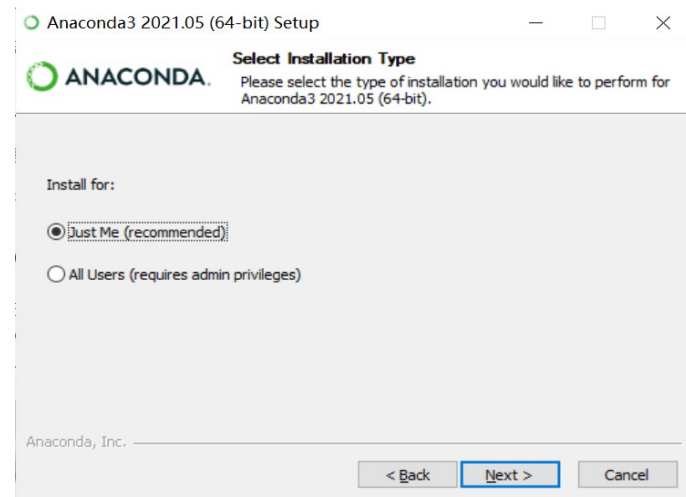


图 1-2-3 选择安装类型界面

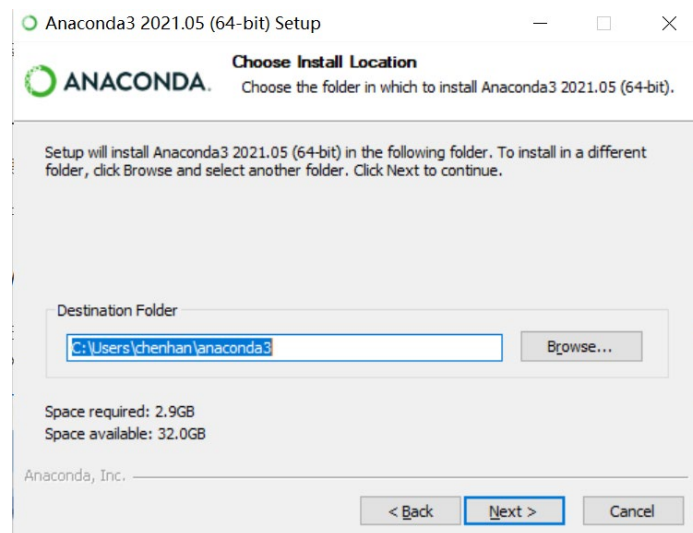


图 1-2-4 选择安装地址界面

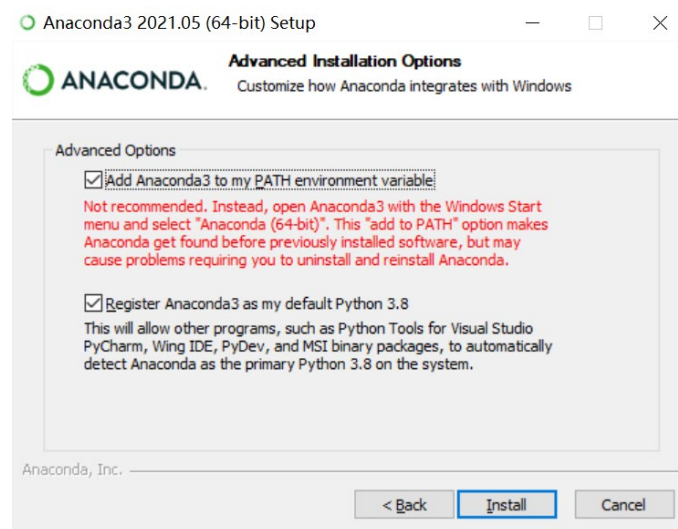


图 1-2-5 高级安装选项界面

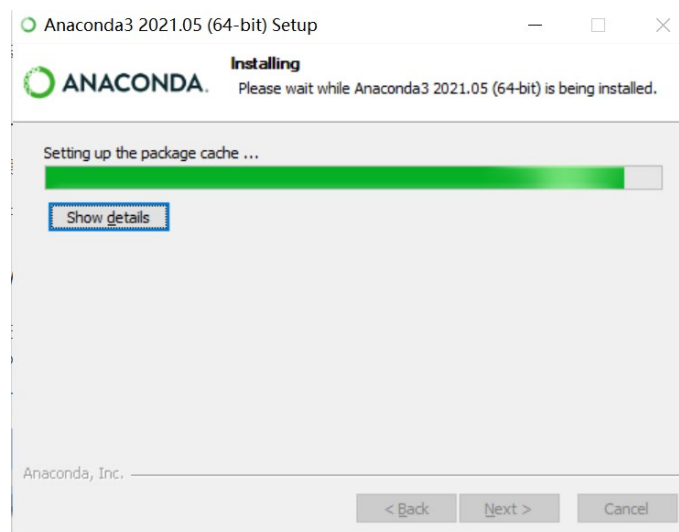


图 1-2-6 安装等待界面

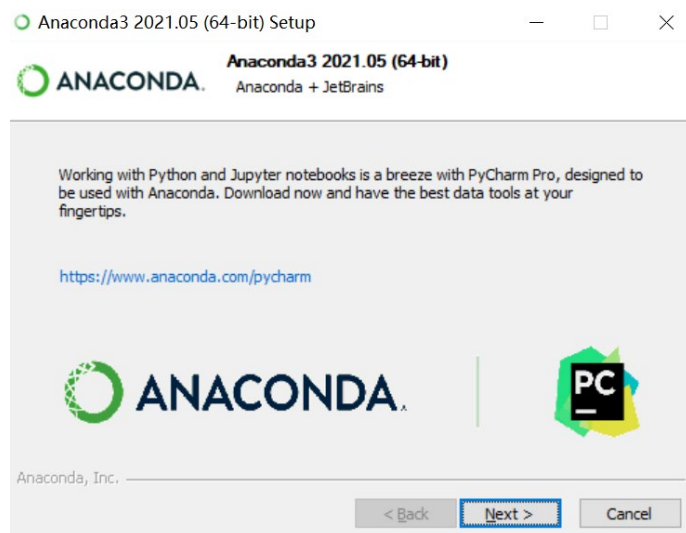


图 1-2-7 Setup 界面

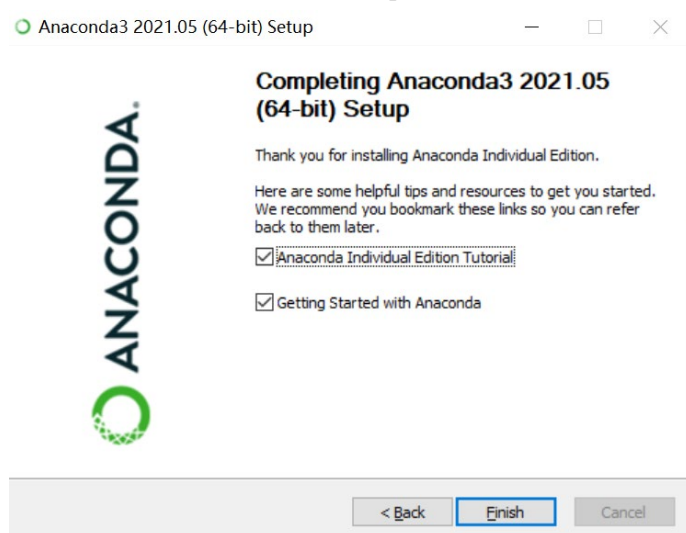


图 1-2-8 安装结束界面

安装完成后，可在命令行检查是否可以使用普通的 Python 环境（直接安装法）或者 Anaconda 虚拟环境（Anaconda 安装法）。（Windows 进入命令行的方式，用户可在电脑内任意文件夹内或者桌面等空白处，按住 Shift 并点击鼠标右键，选择“在此处打开 Powershell 窗口”即可进入命令行窗口）在 Powershell 内输入 Python。如图 1-3 所示为普通的 Python 环境，版本为 3.8.0；图 1-4 提示已启用 Anaconda 的虚拟 base 环境，Python 版本为 3.8.8。此时可以在此处编写 Python 语句命令，如此处编写 `print('Hello World')`。

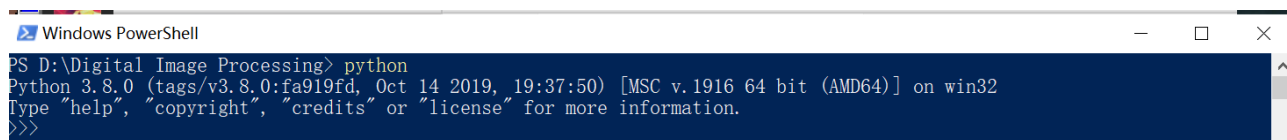
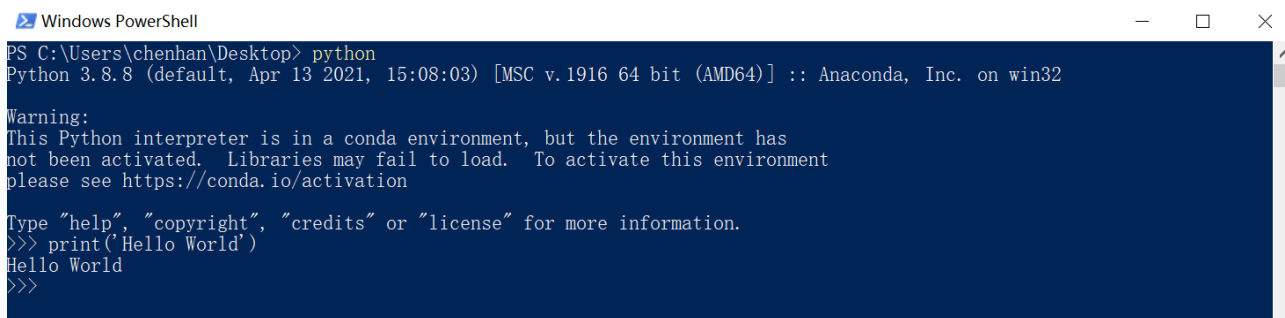


图 1-3 普通 Python 环境的 Powershell 界面



```
PS C:\Users\chenhan\Desktop> python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>>
```

图 1-4 Anaconda 虚拟环境的 Powershell 界面

上述两种安装方式选择一种安装完成后，可以选择安装编辑器用于编写 Python 程序。例如 Notepad++（较为推荐，简单，轻量级。<https://notepad-plus-plus.org/downloads/>），在官网下载 64 Bit 的 Installer，即可以下载对应的 exe 文件，双击按照提示安装即可；或者安装 PyCharm，选择 Community 版本（<https://www.jetbrains.com/pycharm/download/#section=windows>）；或者 Anaconda 中 Spyder 也可以。

接下来演示使用 Notepad++编写第一个 Python 程序，并运行该程序。首先打开 Notepad++文本编辑器，点击文件，选择新建，在文本编辑区域开始编写 `print('Hello World')`；随后再点开文件，选择另存为，选择对应的文件夹位置保存程序代码。保存的文件名要以 .py 作为后缀名，例如这里保存为 `test.py`。文件保存如图 1-5 所示，在程序保存对应的路径处（如此处为 `D:\Digital Image Processing`）打开 Powershell 运行该程序的结果如图 1-6 所示。

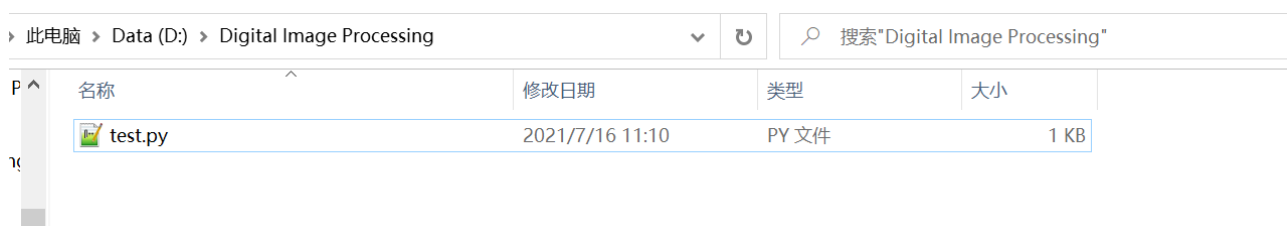


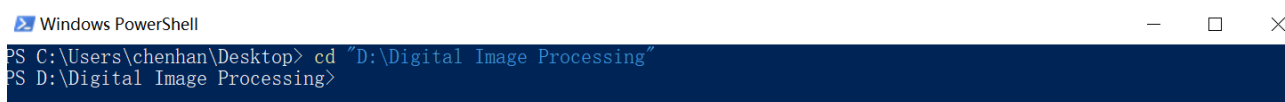
图 1-5 文件存放界面



```
PS D:\Digital Image Processing> python .\test.py
Hello World
PS D:\Digital Image Processing>
```

图 1-6 程序运行结果

第一个 python 程序运行如上图 1-6 所示，已经成功在显示器处打印线上'Hello world'；要注意 Powershell 此时显示的路径是 `D:\Digital Image Processing`，这是存放程序的路径，如果路径不符合，可以使用 `cd "D:\Digital Image Processing"`进入对应的路径，图 1-7 所示为从桌面路径进入 `Digital Image Processing` 文件夹路径，随后在执行 `python test.py` 命令即可运行对应的程序。



```
PS C:\Users\chenhan\Desktop> cd "D:\Digital Image Processing"
PS D:\Digital Image Processing>
```

图 1-7 路径变换命令

额外的 python 学习资料可以在网络上自由阅读，如 <https://www.liaoxuefeng.com/wiki/1016959663602400>、<https://www.runoob.com/python3/python3-tutorial.html> 等等。

## 2、Python 第三方图像处理库的安装

Python 中含有较多的第三方图像处理库，我们需要安装 OpenCV (Open Source Computer Vision Library), PIL, scikit-image 共三个第三方库。根据上述的两种不同的安装方式，我们都提供了对应的安装方法。除此之外，一些额外的第三方库如 Numpy、Matplotlib 等安装方法类似，不再赘述。

### 1) 直接安装:

**OpenCV:** 我们可以选择打开 Powershell，直接输入 `pip install opencv_python` 即可以成功安装 Open CV 库了；或者我们可以选择从该网站 (<https://www.lfd.uci.edu/~gohlke/pythonlibs/>) 下载 OpenCV 的 whl 文件 (`opencv_python-4.5.3-cp38-cp38-win_amd64.whl`，根据你的 python 版本来下载，此处我们时 python3.8)，随后在下载文件对应的路径打开 Powershell，输入 `pip install opencv_python-4.5.3-cp38-cp38-win_amd64.whl` 即可安装成功。

**PIL:** 我们可以选择打开 Powershell，直接输入 `pip install pillow` 即可以成功安装 PIL 库了；或者我们可以选择从该网站 (<https://www.lfd.uci.edu/~gohlke/pythonlibs/>) 下载 PIL 的 whl 文件 (`Pillow-8.3.1-cp38-cp38-win_amd64.whl`，根据你的 python 版本来下载，此处我们时 python3.8)，随后在下载文件对应的路径打开 Powershell，输入 `pip install Pillow-8.3.1-cp38-cp38-win_amd64.whl` 即可安装成功。

**scikit-image:** 我们可以选择打开 Powershell，直接输入 `pip install scikit-image` 即可以成功安装 scikit-image 库了；或者我们可以选择从该网站 (<https://www.lfd.uci.edu/~gohlke/pythonlibs/>) 下载 scikit-image 的 whl 文件 (`scikit_image-0.18.2-cp38-cp38-win_amd64.whl`，根据你的 python 版本来下载，此处我们时 python3.8)，随后在下载文件对应的路径打开 Powershell，输入 `pip install scikit_image-0.18.2-cp38-cp38-win_amd64.whl` 即可安装成功。

**2) Anaconda 安装:** 首先打开 Anaconda 软件，随后点击 Environments，选择所需的虚拟环境，如点击 base (root) 旁边的绿色按键，选择 Open Terminal。在命令行处输入以下对应的命令即可以安装第三方库了。

**OpenCV:** 直接输入 `pip install opencv_python` 即可以成功安装 OpenCV 库了。

**PIL:** 直接输入 `pip install pillow` 即可以成功安装 PIL 库了。

**scikit-image:** 直接输入 `pip install scikit-image` 即可以成功安装 scikit-image 库了。

## 3、Python 语言的基本操作

Python 是一种面向对象的解释型高级程序设计语言，其语法简洁、清晰、优雅，能够使初学者尽快从语法规则中走出来，从而更加注重于解决问题方法的研究。Python 语言具有大量优秀的第三方函数模块，能够满足绝大多数应用领域的开发需求。目前，基于 Python 的相关技术正在飞速发展，用户的数量也在急速增长。

掌握基础的 Python 语法知识，可以重点掌握以下几个内容：数据类型、变量与常量、运算符、选择与循环、列表与元组、字典、函数等。

## 4、阅读 Python 第三方库的文档教程

以下是一些本课程中使用频率较高的第三方库的文档教程，或是教学网站，尝试阅读里面的文档教程。在实际使用过程中，遇到需要使用相关的函数时，可以使用这些文档辅助编程以实现相应功能。

**OpenCV:** [https://docs.opencv.org/master/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/master/d6/d00/tutorial_py_root.html) (官方)、<http://www.woshicver.com/> (中文文档，翻译版)、<http://www.lzlab.com/wiki/python-opencv-tutorial/> (入门教程)

**PIL:** <https://pillow.readthedocs.io/en/stable/handbook/tutorial.html>

**scikit-image:** [https://scikit-image.org/docs/stable/user\\_guide.html](https://scikit-image.org/docs/stable/user_guide.html) (User Guide)、<https://scikit-image.org>

[g/docs/dev/auto\\_examples/](https://docs.dev.auto_examples/) (Example)、<https://scikit-image.org/docs/stable/api/api.html> (API)、<https://scikit-image.org/docs/stable/> (Docs)、<https://cloud.tencent.com/developer/doc/1221> (中文)

**Numpy:** <https://numpy.org/> (官网)、<https://www.runoob.com/numpy/numpy-tutorial.html> (教程)

**Matplotlib:** <https://matplotlib.org/> (官网)、<https://www.runoob.com/matplotlib/matplotlib-tutorial.html> (教程)

## 5、Python 中图像数据与属性信息的获取

**OpenCV:** 使用 `cv2.imread` 函数便可以读入一张图片，查阅其 Numpy 的 `ndarray` 数据格式的属性便可以查看该图片的属性信息。详细的例子如下所示：

```
# -*- coding: utf-8 -*-
import numpy as np
import cv2

# 导入一张图像 模式为彩色图片
img = cv2.imread('images/input.jpg', cv2.IMREAD_COLOR)

print("=====打印一下图像的属性=====")
print("图像对象的类型 {}".format(type(img)))
print(img.shape)
print("图像宽度: {} pixels".format(img.shape[1]))
print("图像高度: {} pixels".format(img.shape[0]))
print("通道: {}".format(img.shape[2]))
print("图像分辨率: {}".format(img.size))
print("数据类型: {}".format(img.dtype))
```

**PIL:** 使用 `Image` 模块中的 `open()` 函数便可以读入一张图片，通过其实例属性便可以查看其图片对应属性信息。详细的例子如下所示：

```
# -*- coding: utf-8 -*-
from PIL import Image

# 导入一张图像
img = Image.open('images/input.jpg')

print("=====打印一下图像的属性=====")
print("图像宽度: {} pixels".format(img.size[0]))
print("图像高度: {} pixels".format(img.size[1]))
print("图像分辨率: {}".format(img.size))
print("图像的模式: {}".format(img.mode))
print("图像的格式: {}".format(img.format))
```



**scikit-image:** 使用 `io` 模块中的 `imread()` 函数便可以读入一张图片，查阅其 Numpy 的 `ndarray` 数据格式的属性便可以查看该图片的属性信息。详细的例子如下所示：

```
# -*- coding: utf-8 -*-
from skimage import io

# 导入一张图像
img = io.imread('images/input.jpg')

print("=====打印一下图像的属性=====")
print("图像对象的类型 {}".format(type(img)))
print(img.shape)
print("图像宽度: {} pixels".format(img.shape[1]))
print("图像高度: {} pixels".format(img.shape[0]))
print("通道: {}".format(img.shape[2]))
print("图像分辨率: {}".format(img.size))
print("数据类型: {}".format(img.dtype))
```

## 6、图像文件的显示

Matplotlib 是 Python 中常用的绘图库，它能让使用者很轻松地将数据图形化，并且提供多样化的输出格式。Matplotlib 可以用来绘制各种静态，动态，交互式的图表。在本课程中，我们使用该库来显示一些图像，以及展示绘制的一些折线图等。详细的例子如下所示，同时思考为何 OpenCV 读入的图像显示为何会出错，如何可以正常显示 OpenCV 读入的图像呢？

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import cv2
from PIL import Image
from skimage import io

# 导入一张图像
img_cv = cv2.imread('images/input.jpg', cv2.IMREAD_COLOR)
img_pil = Image.open('images/input.jpg')
img_sk = io.imread('images/input.jpg')

# 创建小图 1
# 将小图分成 1 行 3 列,第三个参数表示第 n 个图
subplot = plt.subplot(1, 3, 1)
plt.imshow(img_cv)
subplot.set_title('OpenCV') # 设置 title
subplot = plt.subplot(1, 3, 2)
plt.imshow(img_pil)
subplot.set_title('PIL') # 设置 title
```



```
subplot = plt.subplot(1, 3, 3)
subplot.set_title('scikit-image') # 设置 title
plt.imshow(img_sk)

plt.show()
```

## 7、图像类型的转换

图像可以以不同的文件格式和不同的模式（类型）保存，其中图像文件流行的格式包括 BMP（8 位、24 位、32 位）、PNG、JPG（JPEG）、GIF、PPM 和 TIFF。Python 图像处理库可以读取图像，并且提取数据和其他的一些有用信息（例如图像尺寸、类型/模式和数据类型）。

这里主要介绍如何使用 OpenCV 读取一种文件格式的图像并且将其保存为另一种文件格式，将 PNG 格式的图像保存为 JPG 格式的图像。详细的例子如下所示：

```
# -*- coding: utf-8 -*-
import numpy as np
import cv2

# 导入一张图像 模式为彩色图片
img = cv2.imread('images/input.png', cv2.IMREAD_COLOR)
cv2.imwrite('images/input.jpg', img)
```

## 8、彩色图像的灰度化

彩色图像的灰度化，主要包括最大值灰度化、平均值灰度化以及加权平均灰度化等方法。我们定义  $f(x, y)$  为表示位于空间位置  $(x, y)$  处的像素（该像素的红色（R）分量、绿色（G）分量、蓝色（B）分量值分别为  $R(x, y)$ 、 $G(x, y)$ 、 $B(x, y)$ ）的灰度化结果。其中最大值灰度化方法如公式(1)所示，平均值灰度化方法如公式(2)所示，加权平均灰度化方法如公式(3)所示。

$$(1) \quad f(x, y) = \max(R(x, y), G(x, y), B(x, y)) \quad (1)$$

$$(2) \quad f(x, y) = (R(x, y) + G(x, y) + B(x, y)) \div 3 \quad (2)$$

$$(3) \quad f(x, y) = 0.3 \times R(x, y) + 0.59 \times G(x, y) + 0.11 \times B(x, y) \quad (3)$$