



计算机视觉

彭盛霖

西北大学信息学院

pengshenglin@nwu.edu.cn

Python图像处理入门

编译型语言和解释型语言



编译

- 高级语言编写的程序称为源代码（Source Code）或源文件。
- 编译（Compile）是将源代码全部翻译成机器指令，再加上一些描述信息，生成一个新的文件。这个新的文件称为可执行文件。
- 一个可执行文件可以在计算机上多次运行，因此，在不修改源代码的情况下，只需要做一次编译即可。

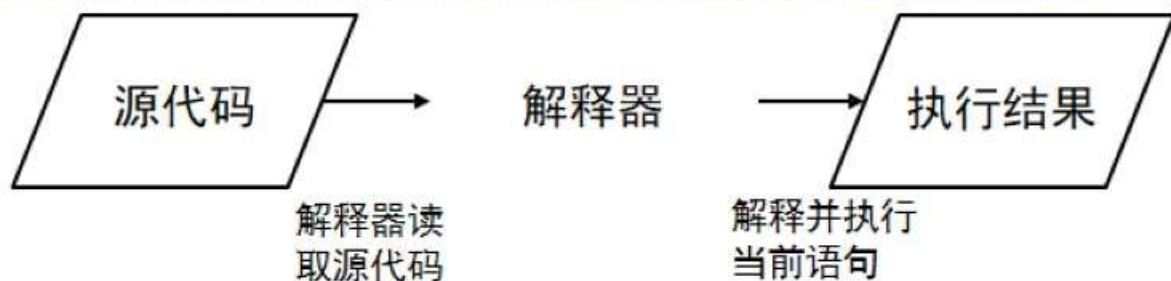


编译型语言和解释型语言



解释

- 解释 (Interpret) 是在程序运行时才对源代码进行逐条语句的翻译 并运行。
- 解释型语言编写的程序每执行一次，就要翻译一次，翻译结果不会像编译型语言一样保存在可执行文件中，因此**效率较低**。



提示



Python是一种解释型语言，但为了提高运行效率，Python程序在执行一次之后会自动生成扩展名为.pyc的字节码文件；字节码不同于机器语言，但很容易转换为机器语言。

1 Python图像处理入门-实战部分



安装包下载

环境安装



清华大学开源软件镜像站

HOME EVENTS BLOG RSS PODCAST MIRRORS

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda>

Index of /anaconda/miniconda/

Last Update: 2022-08-30 14:15

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
Miniconda3-py39_4.12.0-MacOSX-arm64.sh	52.2 MiB	2022-06-02 03:47
Miniconda3-py39_4.12.0-MacOSX-arm64.pkg	63.5 MiB	2022-06-02 03:47
Miniconda3-py38_4.12.0-MacOSX-arm64.sh	52.5 MiB	2022-06-02 03:47
Miniconda3-py38_4.12.0-MacOSX-arm64.pkg	63.8 MiB	2022-06-02 03:47
Miniconda3-latest-MacOSX-arm64.pkg	63.5 MiB	2022-06-02 03:47
Miniconda3-latest-MacOSX-arm64.sh	52.2 MiB	2022-06-02 03:47
Miniconda3-py39_4.12.0-Windows-x86_64.exe	71.2 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-Windows-x86.exe	67.8 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-MacOSX-x86_64.pkg	62.7 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-MacOSX-x86_64.sh	56.0 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-Linux-s390x.sh	69.2 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-Linux-x86_64.sh	73.1 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-Linux-aarch64.sh	75.3 MiB	2022-05-17 04:02
Miniconda3-py39_4.12.0-Linux-ppc64le.sh	74.3 MiB	2022-05-17 04:02
Miniconda3-py38_4.12.0-Windows-x86.exe	66.8 MiB	2022-05-17 04:02
Miniconda3-py38_4.12.0-Windows-x86_64.exe	70.6 MiB	2022-05-17 04:02
Miniconda3-py38_4.12.0-MacOSX-x86_64.sh	56.4 MiB	2022-05-17 04:02



1 Python图像处理入门-实战部分

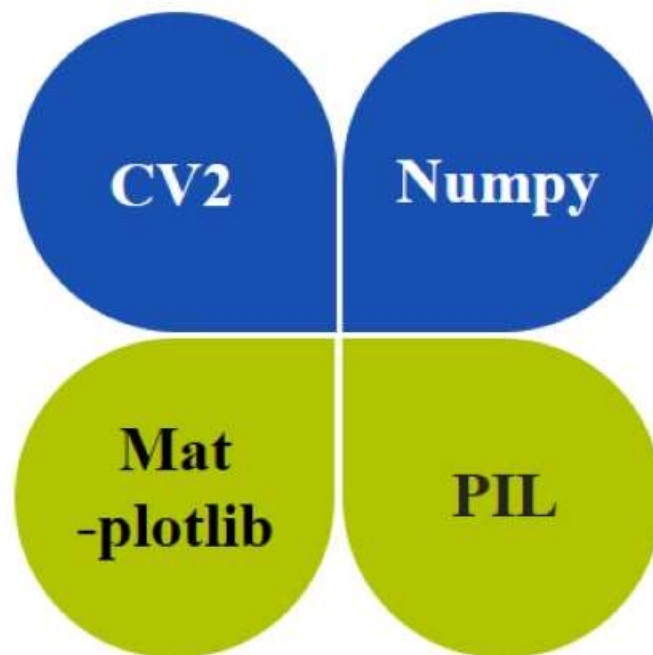
环境安装



图像处理模块安装

```
conda update conda  
conda install spyder  
conda install matplotlib  
conda install opencv  
conda install pillow
```

```
conda update -all  
conda clean -y --all
```



1 Python图像处理入门-实战部分



图像处理模块安装

环境安装

conda list

管理员: Anaconda Prompt

```
(base) C:\Users\Administrator>conda list
# packages in environment at D:\Pytorch:
#
# Name                        Version           Build    Channel
_pytorch_select               0.1               cpu_0    defaults
abs1-py                       1.0.0             pyhd8edlab_0  https://mirror
alabaster                     0.7.12            py_0      defaults
argh                           0.26.2            py38_0     defaults
argon2-cffi                   20.1.0             py38he774522_1  defaults
astroid                       2.4.2             py38_0     defaults
async_generator               1.10              py_0      defaults
atomicwrites                  1.4.0             py_0      defaults
attrs                         20.2.0            py_0      defaults
autopep8                     1.5.4             py_0      defaults
babel                         2.8.0             py_0      defaults
backcall                      0.2.0             py_0      defaults
bcrypt                        3.2.0             py38he774522_0  defaults
```

matplotlib
numpy
opencv
pillow
spyder



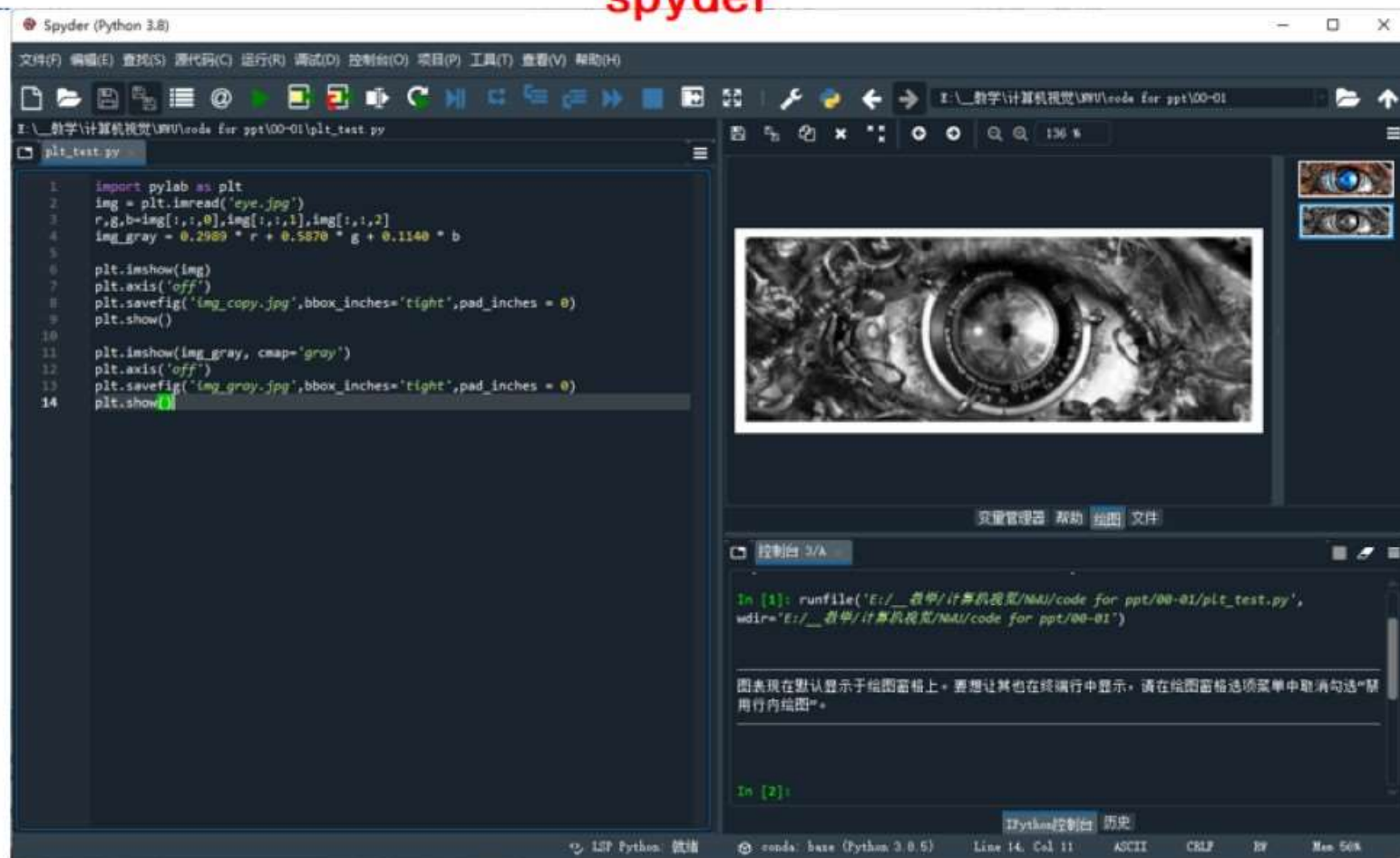
1 Python图像处理入门-实战部分



集成开发环境

环境安装

spyder



1 Python图像处理入门-实战部分

HelloWorld程序



HelloWorld.py

```
1  def main():  
2      # 这是单行注释.....  
3      '''  
4          This is a Hello World Python program  
5          Author: Shenglin Peng  
6          Create Date: 08/30/2022  
7          '''  
8      print("你好, 世界!")  
9  
10  
11  if __name__ == '__main__':  
12      import time  
13      since = time.time()  
14      main()  
15      time_elapsed = time.time() - since  
16      print('The code run %.6f seconds.' % (time_elapsed))
```

注释
缩进
函数



1 Python图像处理入门-实战部分



HelloWorld程序

The screenshot shows a Jupyter Notebook window titled '控制台 4/A'. The top bar contains tabs for '变量管理器', '帮助', '绘图', and '文件'. The main area displays the output of a Python script execution:

```
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.  
  
IPython 7.18.1 -- An enhanced Interactive Python.  
  
In [1]: runfile('E:/__教学/计算机视觉/NWU/code for ppt/00-01/HelloWorld.py', w  
你好，世界！  
The code run 0.000000 seconds.  
  
In [2]:
```



Python程序的调试

- 参考出错提示，`print`调试信息。
- 逐条语句执行程序查看变量状态。
- 设置断点让程序执行时遇到断点就暂停执行、停在断点所在的代码处。
- 目前编写的程序都比较简单，不容易出现逻辑错误；编写复杂程序时如果遇到逻辑错误，可参考网上材料尝试通过调试解决问题。



1 Python图像处理入门-实战部分

图片加载、灰度化、显示和保存



PIL_test.py x

```
1 from PIL import Image
2
3 img = Image.open('eye.jpg')
4 imgGray = img.convert('L')
5
6 img.show()
7 imgGray.show()
8
9 img.save('img_copy.jpg')
10 imgGray.save('img_gray.jpg')
```



PIL



1 Python图像处理入门-实战部分

图片加载、灰度化、显示和保存

```
import cv2
img = cv2.imread('eye.jpg')
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imshow("img",img)
while cv2.waitKey(10)!= 27: # loop if not get ESC
    if cv2.getWindowProperty('img',cv2.WND_PROP_VISIBLE) <= 0:break
cv2.destroyAllWindows()

cv2.imshow("img",img_gray)
while cv2.waitKey(10)!= 27: # loop if not get ESC
    if cv2.getWindowProperty('img',cv2.WND_PROP_VISIBLE) <= 0:br
cv2.destroyAllWindows()

cv2.imwrite('img_copy.jpg', img)
cv2.imwrite('img_gray.jpg', img_gray)
```



CV2



1.2 Python图像处理入门-实战部分

图片加载、灰度化、显示和保存

```
plt_test.py x
1 import pylab as plt
2 img = plt.imread('eye.jpg')
3 r,g,b=img[:, :, 0],img[:, :, 1],img[:, :, 2]
4 img_gray = 0.2989 * r + 0.5870 * g + 0.1140 * b
5
6 plt.imshow(img)
7 plt.axis('off')
8 plt.savefig('img_copy.jpg',bbox_inches=
9 plt.show()
10
11 plt.imshow(img_gray, cmap='gray')
12 plt.axis('off')
13 plt.savefig('img_gray.jpg',bbox_inches='tight',pad_inches = 0)
14 plt.show()
```

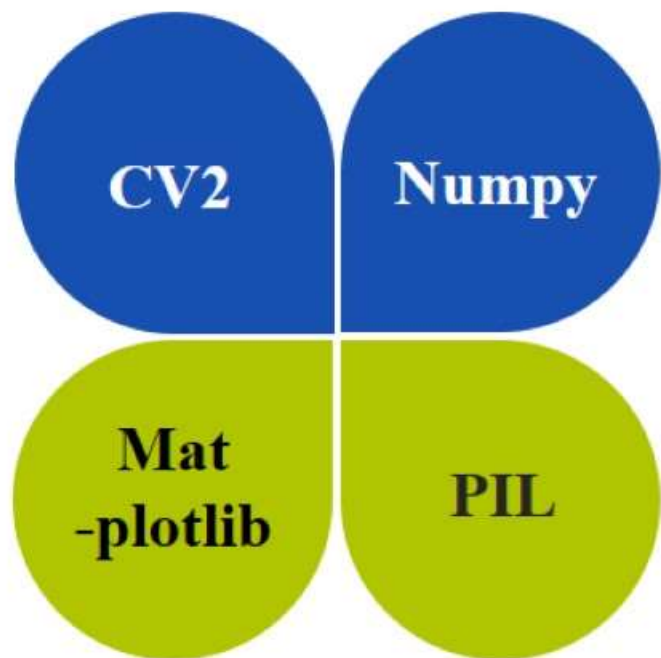


Mat
-plotlib



Python图像处理-实战部分

◆ 思考&实践



➤ 善用百度、谷歌

- 各模块读取的图片都是什么格式

`print(type(img))`

- 各图像格式如何转化

`numpy`

- CV2、PIL模块有哪些内置的图像处理方法

1 Python图像处理入门-实战部分

◆ 接下来的时间：上机实验并完成实验报告

· 《计算机视觉》实验报告

· 实验 01: Python 图像处理入门

姓名		学号	
实验地点		实验日期	

一、实验内容

【1】熟悉并熟练进行 Python 安装和环境配置，完成 windows 环境下 Python 的安装。

【2】熟悉并熟练使用 matplotlib、PIL、cv2 读取、显示和保存图片。

【3】熟悉格式转换，即用一种模块读取图片，用另一种模块保存。

【4】熟悉基本的图像处理函数，实现图像的灰度化、二值化、缩放、旋转、裁剪等基本操作。

