《计算机视觉》实验报告

# 实验01：Python图像处理入门

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 庞晓宇 | 学号 | 2020118100 |
| 实验地点 | 信息学院310 | 实验日期 | 2023年9月1日 |

**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。

1. **实验过程以及结果分析**

Python安装和环境配置，根据向导来就行了。

使用matplotlib、PIL、cv2读取、显示和保存图片。

from PIL import Image  
  
img = Image.open("eye.jpg")  
imgGray = img.convert("L")  
  
img.show()  
imgGray.show()  
  
img.save("img\_copy.jpg")  
imgGray.save("img\_gray.jpg")  
  
# --------------------------------------------------  
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:  
 break  
cv2.destroyAllWindows()  
  
cv2.imwrite("img\_copy.jpg", img)  
cv2.imwrite("img\_gray.jpg", img\_gray)  
  
  
# --------------------------------------------------  
import pylab as plt  
  
img = plt.imread("eye.jpg")  
r, g, b = img[:, :, 0], img[:, :, 1], img[:, :, 2]  
img\_gray = 0.2989 \* r + 0.5870 \* g + 0.1140 \* b  
  
plt.imshow(img)  
plt.axis("off")  
plt.savefig("img\_copy.jpg", bbox\_inches="tight", pad\_inches=0)  
plt.show()  
  
plt.imshow(img\_gray, cmap="gray")  
plt.axis("off")  
plt.savefig("img\_gray.jpg", bbox\_inches="tight", pad\_inches=0)  
plt.show()

熟悉格式转换，即用一种模块读取图片，用另一种模块保存。

import cv2  
import matplotlib.pyplot as plt  
import numpy as np  
from PIL import Image  
  
  
def read\_image(file\_path, library):  
 if library.lower() == "matplotlib":  
 img = plt.imread(file\_path)  
 elif library.lower() == "pil":  
 img = Image.open(file\_path)  
 elif library.lower() == "cv2":  
 img = cv2.imread(file\_path)  
 else:  
 raise ValueError("Invalid library name")  
 return img  
  
  
def matplotlib2PIL(img):  
 return Image.fromarray((img \* 255).astype("uint8"))  
  
  
def PIL2matplotlib(img):  
 return np.asarray(img) / 255  
  
  
def matplotlib2cv2(img):  
 return (img \* 255).astype("uint8")  
  
  
def cv22matplotlib(img):  
 return img / 255  
  
  
def PIL2cv2(img):  
 return np.asarray(img)  
  
  
def cv22PIL(img):  
 return Image.fromarray(img)  
  
  
def convert\_image(img, from\_library, to\_library):  
 if from\_library.lower() == "matplotlib" and to\_library.lower() == "pil":  
 return matplotlib2PIL(img)  
 elif from\_library.lower() == "pil" and to\_library.lower() == "matplotlib":  
 return PIL2matplotlib(img)  
 elif from\_library.lower() == "matplotlib" and to\_library.lower() == "cv2":  
 return matplotlib2cv2(img)  
 elif from\_library.lower() == "cv2" and to\_library.lower() == "matplotlib":  
 return cv22matplotlib(img)  
 elif from\_library.lower() == "pil" and to\_library.lower() == "cv2":  
 return PIL2cv2(img)  
 elif from\_library.lower() == "cv2" and to\_library.lower() == "pil":  
 return cv22PIL(img)  
 else:  
 raise ValueError("Invalid library name")  
  
  
def save\_image(img, file\_path, library):  
 if library.lower() == "matplotlib":  
 plt.imsave(file\_path, img)  
 elif library.lower() == "pil":  
 img.save(file\_path)  
 elif library.lower() == "cv2":  
 cv2.imwrite(file\_path, img)  
 else:  
 raise ValueError("Invalid library name")  
  
  
def show\_image(img, library):  
 if library.lower() == "matplotlib":  
 plt.imshow(img)  
 plt.axis("off")  
 plt.show()  
 elif library.lower() == "pil":  
 img.show()  
 elif library.lower() == "cv2":  
 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()  
 else:  
 raise ValueError("Invalid library name")  
  
  
# test  
if \_\_name\_\_ == "\_\_main\_\_":  
 # 测试图像读取  
 img\_cv2 = read\_image("eye.jpg", "cv2")  
 print(type(img\_cv2)) # <class 'numpy.ndarray'>  
  
 img\_pil = read\_image("eye.jpg", "pil")  
 print(type(img\_pil)) # <class 'PIL.JpegImagePlugin.JpegImageFile'>  
  
 img\_matplotlib = read\_image("eye.jpg", "matplotlib")  
 print(type(img\_matplotlib)) # <class 'numpy.ndarray'>  
 print()  
  
 # 测试图像转换  
 img\_cv2\_to\_pil = convert\_image(img\_cv2, "cv2", "pil")  
 print(type(img\_cv2\_to\_pil)) # <class 'PIL.Image.Image'>  
  
 img\_pil\_to\_cv2 = convert\_image(img\_pil, "pil", "cv2")  
 print(type(img\_pil\_to\_cv2)) # <class 'numpy.ndarray'>  
  
 img\_matplotlib\_to\_pil = convert\_image(img\_matplotlib, "matplotlib", "pil")  
 print(type(img\_matplotlib\_to\_pil)) # <class 'PIL.Image.Image'>  
  
 img\_pil\_to\_matplotlib = convert\_image(img\_pil, "pil", "matplotlib")  
 print(type(img\_pil\_to\_matplotlib)) # <class 'numpy.ndarray'>  
  
 img\_cv2\_to\_matplotlib = convert\_image(img\_cv2, "cv2", "matplotlib")  
 print(type(img\_cv2\_to\_matplotlib)) # <class 'numpy.ndarray'>  
  
 img\_matplotlib\_to\_cv2 = convert\_image(img\_matplotlib, "matplotlib", "cv2")  
 print(type(img\_matplotlib\_to\_cv2)) # <class 'numpy.ndarray'>  
  
 # 测试图像显示  
 show\_image(img\_cv2, "cv2")  
 show\_image(img\_pil, "pil")  
 show\_image(img\_matplotlib, "matplotlib")  
  
 # 测试图像保存  
 save\_image(img\_cv2, "eye\_cv2.jpg", "cv2")  
 save\_image(img\_pil, "eye\_pil.jpg", "pil")  
 save\_image(img\_matplotlib, "eye\_matplotlib.jpg", "matplotlib")

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。

import cv2  
from PIL import Image  
  
  
def test\_cv2\_grayscale():  
 # 加载图像  
 img = cv2.imread("eye.jpg")  
  
 # 测试灰度化方法  
 gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
 assert gray.shape == (img.shape[0], img.shape[1])  
  
  
def test\_cv2\_threshold():  
 # 加载图像  
 img = cv2.imread("eye.jpg")  
  
 # 测试阈值化方法  
 gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
 \_, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH\_BINARY)  
 assert thresh.shape == (img.shape[0], img.shape[1])  
  
  
def test\_cv2\_resize():  
 # 加载图像  
 img = cv2.imread("eye.jpg")  
  
 # 测试缩放方法  
 resized = cv2.resize(img, (100, 100))  
 assert resized.shape == (100, 100, 3)  
  
  
def test\_cv2\_rotate():  
 # 加载图像  
 img = cv2.imread("eye.jpg")  
  
 # 测试旋转方法  
 rows, cols, \_ = img.shape  
 M = cv2.getRotationMatrix2D((cols / 2, rows / 2), 45, 1)  
 rotated = cv2.warpAffine(img, M, (cols, rows))  
 assert rotated.shape == (rows, cols, 3)  
  
  
def test\_cv2\_crop():  
 # 加载图像  
 img = cv2.imread("eye.jpg")  
  
 # 测试裁剪方法  
 rows, cols, \_ = img.shape  
 cropped = img[rows // 4 : rows // 2, cols // 4 : cols // 2]  
 assert cropped.shape == (rows // 4, cols // 4, 3)  
  
  
def test\_pil\_grayscale():  
 # 加载图像  
 img = Image.open("eye.jpg")  
  
 # 测试灰度化方法  
 gray = img.convert("L")  
 assert gray.size == img.size  
  
  
def test\_pil\_threshold():  
 # 加载图像  
 img = Image.open("eye.jpg")  
  
 # 测试阈值化方法  
 gray = img.convert("L")  
 thresh = gray.point(lambda x: 255 if x > 127 else 0)  
 assert thresh.size == img.size  
  
  
def test\_pil\_resize():  
 # 加载图像  
 img = Image.open("eye.jpg")  
  
 # 测试缩放方法  
 resized = img.resize((100, 100))  
 assert resized.size == (100, 100)  
  
  
def test\_pil\_rotate():  
 # 加载图像  
 img = Image.open("eye.jpg")  
  
 # 测试旋转方法  
 rotated = img.rotate(45)  
 assert rotated.size == img.size  
  
  
def test\_pil\_crop():  
 # 加载图像  
 img = Image.open("eye.jpg")  
  
 # 测试裁剪方法  
 cropped = img.crop(  
 (img.width // 4, img.height // 4, img.width // 2, img.height // 2)  
 )  
 assert cropped.size == (img.width // 4, img.height // 4)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 test\_cv2\_grayscale()  
 test\_cv2\_threshold()  
 test\_cv2\_resize()  
 test\_cv2\_rotate()  
 test\_cv2\_crop()  
 test\_pil\_grayscale()  
 test\_pil\_threshold()  
 test\_pil\_resize()  
 test\_pil\_rotate()  
 test\_pil\_crop()  
 print("所有测试通过！")

1. **实验总结**

在本次实验中，我熟悉并掌握了Python的安装和环境配置、依赖的下载，以及使用matplotlib、PIL、cv2等模块进行图像的读取、显示和保存；了解了格式转换的相关知识，掌握了基本的图像处理函数，如灰度化、二值化、缩放、旋转、裁剪等操作。这些操作可以通过调用相应的函数来实现，同时也需要注意函数的参数和返回值的类型。

1. **材料提交**

1.实验要求内容完备(实验代码、实验结果及分析)、格式规范、排版美观。

2.实验过程中遇到问题需记录具体问题和解决方法；

3.把相关材料(包括实验报告、实验代码、实验使用到的图片等数据)压缩打包为“计算机视觉实验01\_学号\_姓名.zip”，提交到邮箱pengshenglin@nwu.edu.cn；

4.截止时间为实验课当周周日24点前(如实验课在周六周日，截止时间为下周周二24点前)。

5.不要迟交，不要抄袭(迟交当次作业最多70分，抄袭整个课程记0分！)。实验报告整体雷同且存在以下情况判为抄袭：程序仅有极少字符与变量的不同且；程序仅有空格和分行的不同；存在从网页复制导致的乱码，全角符号，非ASCII符号，&nbsp;等；代码高度相似并且程序存在完全相同的错误。(重复教材上或课件上的代码不计入抄袭)