

# 实践报告

高鹏宇

学号：2302007025

## 一、练习内容与结果

### (一) 命令行环境样例

删除指定目录：

1. `rm -r directory_to_delete`

2. 复制文件到指定目录：

```
cp source_file.txt destination_directory/
```

3. 查看文件内容：

```
cat file.txt
```

4. 查找包含特定文本的文件：

```
grep "search_text" *
```

#### 5. 压缩文件或目录:

```
tar -czvf archive_name.tar.gz directory_to_compress
```

#### 6. 解压文件:

```
tar -xzvf archive_name.tar.gz
```

## (二) Python 入门基础样例

定义函数:

```
1. def greet(name):  
    return f"Hello, {name}!"
```

结果: 定义了一个名为 greet 的函数, 它接受一个参数 name 并返回一个格式化的问候语。

#### 2. 调用函数:

```
print(greet("Alice"))
```

结果: 打印输出 Hello, Alice! 到控制台。

#### 3. 列表操作:

```
fruits = ["apple", "banana", "cherry"]  
fruits.append("orange")
```

结果: 在列表 fruits 中添加了一个元素"orange", 现在 fruits 列表的内容是 ["apple", "banana", "cherry", "orange"]。

#### 4. 字典操作:

```
person = {"name": "Alice", "age": 25}
print(person["name"])
```

结果: 打印输出 Alice 到控制台, 这是字典 person 中键" name" 对应的值。

#### 5. 异常处理:

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
```

结果: 尝试执行除以零的操作, 这会引发一个 ZeroDivisionError 异常。程序捕获这个异常并打印输出 Cannot divide by zero! 到控制台。

#### 6. 文件读写:

```
with open("file.txt", "w") as f:
    f.write("Hello, World!")
```

结果: 创建 (或覆盖) 一个名为 file.txt 的文件, 并向其中写入字符串 Hello, World!。如果文件不存在, 它会被创建; 如果文件已存在, 其内容会被覆盖。

### (三) Python 视觉应用样例

使用 OpenCV 读取图像:

```
1. import cv2
   image = cv2.imread('image.jpg')
```

结果：加载名为 image.jpg 的图像文件到变量 image 中。如果文件不存在或路径错误，image 将是 None。

## 2. 显示图像：

```
cv2.imshow('Image', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

结果：在窗口中显示图像。cv2.waitKey(0) 使得窗口持续打开直到任意键被按下。之后，关闭所有 OpenCV 创建的窗口

## 3. 转换为灰度图像：

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

结果：将 image 转换为灰度图像且保存。

## 4. 保存图像：

```
cv2.imwrite('gray_image.jpg', gray_image)
```

结果：将灰度图像保存。

## 5. 使用 Pillow 打开图像：

```
from PIL import Image
img = Image.open('image.jpg')
```

结果：使用 Pillow 库打开 image.jpg 文件并存储在变量 img 中。

## 6. 调整图像大小：

```
img_resized = img.resize((100, 100))
```

结果：将图像的大小调整为 100x100 像素，并存储在变量中。

7. 绘制图形 (Matplotlib):

```
import matplotlib.pyplot as plt  
plt.plot([1, 2, 3], [4, 5, 6])  
plt.show()
```

结果：在图形界面中绘制一个简单的线图，x 轴为 [1, 2, 3]，y 轴为 [4, 5, 6]。

8. 显示图像 (Pillow):

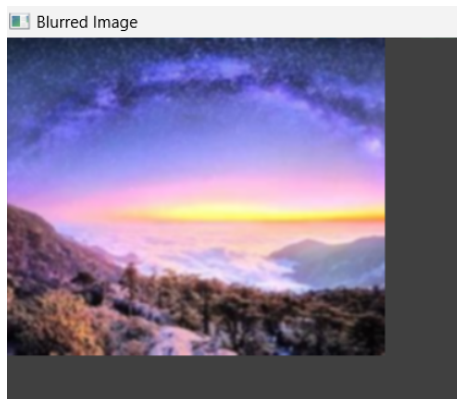
```
img.show()
```

结果：使用默认的图像查看器显示图像 img。

9. 图像滤镜 (Pillow):

```
from PIL import ImageFilter  
img_filtered = img.filter(ImageFilter.BLUR)
```

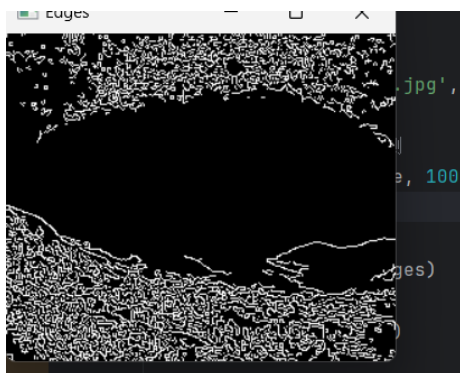
结果：对图像应用模糊滤镜，并将结果存储在变量中。



10. 使用 OpenCV 进行边缘检测：

```
edges = cv2.Canny(image, 100, 200)
cv2.imshow('Edges', edges)
```

结果：使用 Canny 算法对图像 image 进行边缘检测，并将结果存储在 edges 变量中。然后显示边缘检测的结果。



## 二、解题感悟

命令行环境是一个强大的工具，它允许用户通过文本命令直接与操作系统交互。在使用过程中，我发现大多数命令行命令在不同的操作系统（如 Linux、macOS 和 Windows 的特定版本）中都是通用的，这使得跨平台工作变得更加容易。

在使用 Python 的过程中，我发现相较于 C 语言，Python 的语法更简单直观，大量的库和框架，可以用于各种应用，是一种非常泛用的工具。除此之外，得益于像 OpenCV、Pillow 和 Matplotlib 这样的库，Python 在视觉应用方面也非常强大。我们可以使用 OpenCV 和 Pillow，可以轻松进行图像读取、编辑、转换和保存。

总结来说，命令行环境、Python 都是强大的工具，它们提供了广泛的功能和灵活性，可以帮助我们高效地完成任务。我们尽力去学会熟练掌握它们。

GitHub 地址：[https://github.com/Gao-py/class\\_homework](https://github.com/Gao-py/class_homework)