

命令行与 Python

耿惠生 23020007028

September 2024

1 练习内容

1.1 命令行环境

1.1.1 删除空目录

```
rm -rf .
```

1.1.2 移动、重命名

```
mv . .
```

1.1.3 合并

```
cat .
```

1.2 Python 入门基础

1.2.1 循环打印数字

```
for i in range(1, 11):  
    print(i)
```

结果：打印 1 到 10 的数字

1.2.2 求和

```
def add(a, b):  
    return a + b
```

```
result = add(3, 5)
print(result)
```

结果：输出 3 和 5 的和，即 8

1.2.3 读取文件内容

```
with open("file.txt", "r") as file:
    content = file.read()
print(content)
```

结果：打开 file.txt，读取并打印文件内容

1.2.4 利用模板

```
import math
radius = 5
area = math.pi * radius **2
print(area)
```

结果：利用 math，输出圆的面积

1.2.5 斐波那契

```
def fibonacci(n)
if n <= 1:
    return n
else:
    return fibonacci(n-1) + fibonacci(n-2)
n = 10
print("The", n, "th Fibonacci number is:", fibonacci(n))
```

结果：计算 10 的斐波那契数

1.2.6 字典储存

```
student =
    "name":"John",
    "age":20,
    "grade":"A"
```

```
print(student)
```

结果：输出学生信息

1.2.7 阶乘

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial(n-1)
```

```
num = 5
```

```
print("The factorial of", num, "is", factorial(num))
```

结果：输出 5 的阶乘，即 120

1.3 Python 视觉设计

1.3.1 显示图像

```
import cv2
```

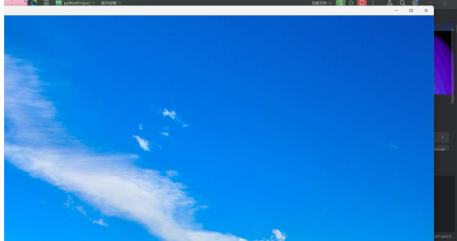
```
image = cv2.imread('example.webp')
```

```
cv2.imshow('Example Image', image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

结果：利用 cv2，读取 example.webp，并显示该图片



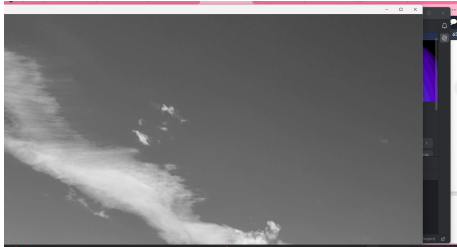
1.3.2 保存图片

```
cv2.imwrite('example.webp', image)
```

结果：保存图片

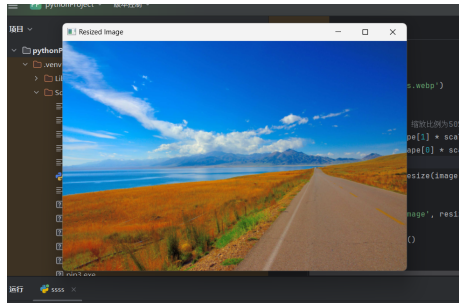
1.3.3 灰化图像

```
import cv2
image = cv2.imread('example.webp')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow('Grayscale Image', gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
结果：彩色图片变为灰色
```



1.3.4 缩放图片

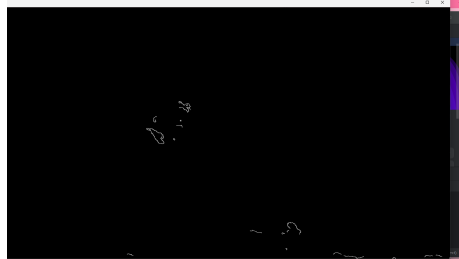
```
import cv2
image = cv2.imread('example.webp')
scale_percent = 50
width = int(image.shape[1] * scale_percent/100)
height = int(image.shape[0] * scale_percent/100)
dim = (width, height)
resized_image = cv2.resize(image, dim, interpolation = cv2.INTER_AREA)
cv2.imshow('Resized Image', resized_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
结果：将图片缩小至 20
```



1.3.5 边缘检测

```
import cv2
image = cv2.imread('example.webp', cv2.IMREAD_GRAYSCALE)
edges = cv2.Canny(image, 100, 200)
cv2.imshow('Edges', edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

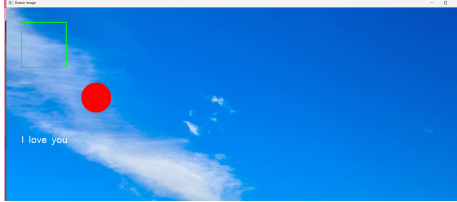
结果：对 example.webp 边缘检测，并输出检测结果



1.3.6 图形绘制与文本插入

```
import cv2
image = cv2.imread('example.webp')
cv2.rectangle(image, (50, 50), (200, 200), (0, 255, 0), 2)
cv2.circle(image, (300, 300), 50, (0, 0, 255), -1)
cv2.putText(image, 'I love you', (50, 450), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
cv2.imshow('Drawn Image', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

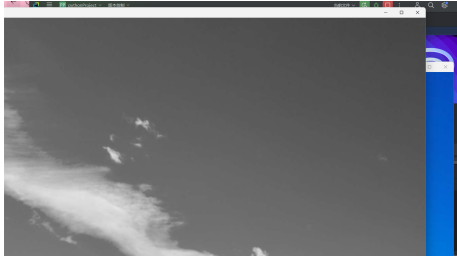
结果：在图片上插入矩形，圆形，并输入文字 “I love you”



1.3.7 高斯模糊

```
blurrediimage = cv2.GaussianBlur(grayiimage, (5, 5), 0)
```

结果：将灰化图片模糊化



1.3.8 轮廓检测

```
import cv2 import numpy as np
image = cv2.imread('example.jpg')
grayiimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurrediimage = cv2.GaussianBlur(grayiimage, (5, 5), 0)
edges = cv2.Canny(blurrediimage, 50, 150)
contours, hierarchy = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(image, contours, -1, (0, 255, 0), 2)
cv2.imshow('Image with Contours', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

结果：绘制图片轮廓并显示轮廓检测后的图片



1.3.9 旋转图片

```
from PIL import Image
def rotate_image(image_path, angle):
    image = Image.open(image_path)
    rotated_image = image.rotate(angle)
    rotated_image.save("rotated_image.jpg")
rotate_image("example.webp", 90)
```

结果: example.webp 旋转了 90 度

2 使用感悟

2.1 Python 视觉设计

2.1.1 简洁

Python 语法简洁, 这使我们在视觉设计中实现复杂的算法和功能, 使用起来十分便利。

2.1.2 库丰富

Python 有许多用于图像处理和视觉设计的库, 如 PIL、OpenCV、Matplotlib 等。这些库提供了丰富的功能, 是我们用简单的代码实现强大的功能。

3 github 地址

<https://github.com/211-sss/homework>