

## 实验 2 处理尺寸不合适和灰度图像

实验难度：一般

实验摘要：对于能够正确读取的图片，需要做进一步处理。首先，判断是否是彩色图还是灰度图，如果是灰度图，不符合应用场景，则移除到 GRAY 目录。然后，判断尺寸，如果尺寸太小（目前任务要求是 200\*200），不合适进一步使用，过滤到 SMALL 目录下。最后,将剩余图片转换成 jpg 格式保存在 DATA 目录下。

实验建议：了解 os 和 opencv 库的相关知识。

实验目标：能够使用 Python 相关库函数完成处理尺寸不合适和灰度图像的任务。

### 1、处理尺寸不合适和灰度图像

- 

#### 1.1、分离彩色图和灰度图

首先，类似的创建保存灰度图像的 GRAY 目录。

```
GRAY_PATH = './GRAY'
```

```
if not os.path.exists(GRAY_PATH):
```

```
    os.mkdir(GRAY_PATH)
```

通过 `img.ndim` 可以读取图像的维度。如果维度为 2, 说明只有一个颜色通道, 是灰度图。

使用 `os.rename` 函数转移到创建好的 GRAY 目录。并通过 `continue` 退出当前循环, 处理下一个文件。

```
ndim = img.ndim
```

```
if ndim == 2:
```

```
    print(filename, " is a gray image")
```

```
    dst_filename = os.path.join(GRAY_PATH, file)
```

```
    os.rename(filename,dst_filename)
```

```
    continue
```

下一步

---

---

- 1.2、分离像素太小的图像

---

- 首先，通过类似的方法，创建保存小尺寸图像的 SMALL 目录。

---

- ```
SMALL_PATH = './SMALL'
```

---

- ```
if not os.path.exists(SMALL_PATH):
```

---

- ```
    os.mkdir(SMALL_PATH)
```

---

- 然后，通过 `img.shape` 可以读取图像的行列像素和深度。注意返回的顺序是图像高度，宽度和深度。

---

- ```
print(file, "size: ", img.shape)
```

---

- ```
h,w,c = img.shape
```

---

- 这里，做一个简单的判断，如果宽度或者高度像素小于 200，则认为图像尺寸过小。通过 `os.rename` 将文件转移到 SMALL 目录，并调用 `continue` 退出当前循环。

---

- ```
if h < 200 or w < 200:
```

---

- ```
    print(filename, " is too small")
```

---

-

```
dst_filename = os.path.join(SMALL_PATH, file)
```

---

- 

```
os.rename(filename, dst_filename)
```

---

- 

```
continue
```

---

- 

- 

- 

下一步

---

- 

- 

- 

- 

### 1.3、统一图像格式

---

- 

经过前面的处理，保留下来的都是合适的图像文件了。重新命名，并保存在 DATA 目录下。

同样，需要先创建 DATA 目录。

---

- 

```
DATA_PATH = './DATA'
```

---

- 

```
if not os.path.exists(DATA_PATH):
```

---

- 

```
os.mkdir(DATA_PATH)
```

---

- 

-

在循环外初始化计数器 cnt, 然后根据计数器创建文件名称。因为文件名类似 000000.jpg, 000001.jpg, 直接用计数器的转成字符串是不行的。所以这里使用到字符串的 format 方法, 通过{:0>6d}用 0 补齐到 6 个字符串长度。

- 

```
#创建类似 000000.jpg, 000001.jpg 的文件名
```

- 

```
newfile = '{:0>6d}'.format(cnt)+".jpg"
```

- 

```
print(newfile)
```

- 

```
cnt += 1
```

- 

- 

最后, 通过 cv2.imwrite 进行文件转换, 并转移到 DATA 目录下使用新名称。

- 

```
dst_filename = os.path.join(DATA_PATH, newfile)
```

- 

```
cv2.imwrite(dst_filename, img)
```

- 

- 

- 

下一步

-



- 
- `if not os.path.exists(GRAY_PATH):`
- `os.mkdir(GRAY_PATH)`
- 
- `SMALL_PATH = './SMALL'`
- 
- `if not os.path.exists(SMALL_PATH):`
- `os.mkdir(SMALL_PATH)`
- 
- `DATA_PATH = './DATA'`
- 
- `if not os.path.exists(DATA_PATH):`
- `os.mkdir(DATA_PATH)`
- 
- 
- `cnt = 0`
- 
- `#遍历文件`
- 
- `for file in os.listdir(ROOT_PATH):`
- `filename = os.path.join(ROOT_PATH, file)`
- 
- `#读取文件`
- 
- `img = cv2.imread(filename, -1)`

- 
- `#如果文件无效，则转移到 REMOVED 目录`

- 
- `if img is None:`

- 
- `print("file: ", file , "is invalid")`

- 
- `dst_filename = os.path.join(REMOVED_PATH, file)`

- 
- `os.rename(filename,dst_filename)`

- 
- `continue`

- 
- 
- `#如果图像是灰度图，则转移到 GRAY 目录`

- 
- `if img.ndim == 2:`

- 
- `print(filename, " is a gray image")`

- 
- `dst_filename = os.path.join(GRAY_PATH, file)`

- 
- `os.rename(filename,dst_filename)`

- 
- `continue`

- 
- `#如果尺寸小于 200*200，则转移到 SMALL 目录`

- 
- `h,w,c = img.shape`



