

理论盘点：人脸识别的一般步骤

目录

- 1 人脸识别的原理
- 2 人脸识别的一般步骤

人脸识别的原理

计算机上的视频（本质也是图片）是由**数字**组成的，因此通过数学计算可以进行图像识别。

人脸识别任务是图像识别的子类，图像识别可拆分为**图像检测**和**图像对比**两个任务：

- 检测图像里的人脸，实际上是分类问题的一种，即**图像里有没有人脸**
- 图像对比是指检测出的人脸和预先采集的人脸的**相似程度**

人脸识别的一般步骤

1. 图片转换为灰度
2. 识别图像中的人脸
3. 训练特征库，将特定的人脸标注并存入特征库
4. 比对目标图片（视频），对目标图像出现的人脸进行判断，根据相似程度进行后续处理

人脸识别的一般步骤

通过 **OpenCV** 库，分析图像的基本组成

- 图像中的每一个像素点由三个元素组成，分别表示 **RGB 三种颜色**
- 一张图片的宽和高由**多个像素点**组成

总结

- 1 人脸识别是指将由特定特征的图像从原始图像中检测出来，并进行处理的工作，人脸识别本质上是分类问题
- 2 一张彩色图像可以由多个像素点组成，像素点包含了 RGB 三种颜色，像素点可以用数字表示，通过数学计算可以进行图像的处理

课后作业

请你使用这节课学习的 OpenCV 库，打开任意一张图片，并输出这张图片的大小信息，输出图片的数字信息。

理论盘点：人脸识别常用的 Python 库有哪些？

目录

- 1 人脸识别的常用库
- 2 使用 OpenCV 读取和保存图片
- 3 使用 OpenCV 标注图像

人脸识别的常用库

人脸识别的常用库有以下几种：

1. 基于特征向量方法：OpenCV
2. 基于 CNN 或 HOG 方法：Tensorflow、PyTorch

使用 OpenCV 读取和保存图片

- 读取图片

`cv2.imread(图片路径和名称)`

- 显示图片

`cv2.imshow()`

- 保存图片

`cv2.imwrite()`

使用 OpenCV 标注图像

标注图像，用于调试时不断优化算法

一般采用**矩形框**对图像进行标注：

`cv.rectangle(图片, 坐标, color=颜色, thickness=宽度)`

总结

- 1 OpenCV 是实现人脸识别的常用库，它比基于深度学习的库更易验证和学习
- 2 OpenCV 能够进行图片的读取、显示、写入以及使用矩形框标注特定区域，在使用其他库进行人脸识别时，也经常配合 OpenCV 进行图像的标注工作

课后作业

请你打开任意一张图片，并使用 `cv.rectangle()` 函数对图像进行标注，标注范围分别为 `50,50,50,50` `100,100,100,100` 标注后保存图片，并查看标注后的图片。

怎样通过 OpenCV 采集视频信号？

目录

- 1 使用 OpenCV 采集图片并转为灰度图片
- 2 使用 OpenCV 采集视频信号并保存图片
- 3 使用 OpenCV 采集 MP4 视频

使用 OpenCV 采集图片并转为灰度图片

转换为灰度是为了更好地提取特征

`cv.cvtColor()` 用于颜色转换

使用 OpenCV 采集视频信号并保存图片

要想识别特定的人脸，需要对 OpenCV 的模型进行训练

训练数据一般是**从摄像头采集的截图**，也可以是事先准备好的图片

- 采集摄像头数据函数：`cv.VideoCapture(0)`
- 从摄像头截图并保存文件函数：`cv.imencode().tofile()`

使用 OpenCV 采集 MP4 视频

`cv.VideoCapture()` 除了捕获摄像头，还可以读取 MP4 格式的文件

通过读取视频可以反复优化 OpenCV 的模型

总结

- 1 OpenCV 能够采集图片和视频中的数据，并对图片的色彩空间做转换
- 2 通过 OpenCV 对视频截图功能，可以构建自己的模型测试图片

课后作业

请你使用OpenCV截取视频中，相同人物至少10张图片，并将图片放入到一个文件夹中

如何训练人脸模型？

目录

1 读取目录下全部的训练图片

2 加载分类模型

3 提取人脸特征值

读取目录下全部训练文件

训练人脸模型的图片，一般会存入到一个文件夹中，往往需要你遍历该目录，将所有文件的路径以字符串格式存放到一个列表中

- 显示目录下所有文件： `os.listdir()`
- 连接目录和文件： `os.path.join()`

加载分类模型

- 调用分类器

```
cv.CascadeClassifier()
```

- 人脸分类器

```
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/  
site-packages/cv2/data/haarcascade_frontalface_default.xml
```

提取人脸特征值

- 图片被转换为数组之后，通过**特征提取函数**将人脸特征值提取出来

人脸特征提取函数: `face_detect.detectMultiScale(图片数组)`

总结

- 1 训练人脸模型的关键步骤是将人脸数据特征与图片进行分离
- 2 通过 OpenCV 训练更多角度的人脸数据，能够让 OpenCV 识别人脸更精确

课后作业

请你使用 `http.server` 模块，将目录 `/tmp` 通过 HTTP 协议发布到 8080 端口，并通过浏览器验证是否发布成功。

怎样设计人脸识别系统？

目录

- 1 调用分类器和训练模型对图片进行人脸识别
- 2 通过标注优化参数

调用分类器和训练模型对图片进行人脸识别

- 读取训练模型函数

`recognizer.read(训练模型)`

- 人脸识别函数

`recognizer.predict()`

通过标注优化参数

默认模型往往识别效果不理想，通过标注人眼分析是否识别准确
并通过检测参数对识别内容进行调整。

总结

- 1 模型数量和参数决定识别准确率
- 2 识别人脸后，可以基于概率进行相应的人脸归属分类

课后作业

请你 OpenCV 实现特定人脸识别功能，识别用户为 A 时，终端输出用户 A，识别出是其他用户时，输出其他用户。

THANKS