# 人脸识别系统教学文档*by 刘海熠*

### 准备工作

在开始之前，确保你的Python环境已经搭建好，并且能够运行基本的Python程序。此外，你需要了解基础的Python语法，比如变量、循环、函数等。

### 第1步：安装必要的库

在终端或命令提示符中运行以下命令来安装所需的库：

-------------------------------------------------------------------------------

***Bash***

pip install facenet-pytorch opencv-python Pillow numpy

-------------------------------------------------------------------------------

这些库的作用如下：

* facenet\_pytorch：用于人脸检测和识别的深度学习模型。
* opencv-python (cv2)：用于图像和视频处理的库。
* Pillow (PIL)：用于图像处理，比如打开、修改和保存图像。
* numpy：用于高效的数学和科学计算。

### 第2步：导入库

在Python脚本的开始部分，你需要导入这些库，以便使用它们提供的功能。

-------------------------------------------------------------------------------

***Python***

from facenet\_pytorch import MTCNN, InceptionResnetV1

import cv2

from PIL import Image

import numpy as np

import os

-------------------------------------------------------------------------------

* from facenet\_pytorch import MTCNN, InceptionResnetV1：从facenet\_pytorch库中导入MTCNN和InceptionResnetV1模型。
* import cv2：导入opencv-python库，通常简称为cv2。
* from PIL import Image：从Pillow库中导入Image模块，用于处理图像。
* import numpy as np：导入numpy库，并通常使用别名np。
* import os：导入os模块，用于处理文件和目录。

#### 第3步：初始化人脸检测和识别模型

在这一步，你将初始化两个关键的模型：MTCNN用于人脸检测，InceptionResnetV1用于人脸特征提取。

-------------------------------------------------------------------------------

***Python***

mtcnn = MTCNN()

resnet = InceptionResnetV1(pretrained='vggface2').eval()

-------------------------------------------------------------------------------

* mtcnn = MTCNN()：创建MTCNN的实例，用于后续的人脸检测。
* resnet = InceptionResnetV1(pretrained='vggface2').eval()：加载预训练的InceptionResnetV1模型（使用vggface2数据集训练），并设置为评估模式。

#### 第4步：创建人脸特征库

你需要创建一个函数，比如load\_and\_extract\_features(folder\_path)，这个函数会遍历指定文件夹中的子文件夹，每个子文件夹代表一个人，并从中提取人脸特征。

* 遍历文件夹，读取每张图像。
* 使用MTCNN检测图像中的人脸。
* 使用InceptionResnetV1模型提取人脸的特征向量。
* 将每个人的特征向量平均，存储在一个字典中，字典的键是人的名字，值是特征向量。

#### 第5步：比较特征向量

实现一个函数compare\_faces(enc1, enc2)，它计算并返回两个特征向量之间的欧氏距离。这个距离越小，表示两个向量越相似。

#### 第6步：人脸识别逻辑

实现recognize\_face函数，该函数接受一张人脸图像，将其特征与特征库中的特征进行比较，找出最匹配的人脸。

* 检测输入图像中的人脸。
* 提取人脸的特征向量。
* 与数据库中的特征进行比较，找到最相似的特征。
* 根据相似度返回最匹配的人名或"unknown"。

#### 第7步：实时视频处理

使用opencv-python捕获视频流，并对每一帧图像进行人脸检测和识别。显示识别结果，并将结果实时绘制在视频帧上。

* 打开摄像头，读取视频帧。
* 对每一帧使用MTCNN进行人脸检测。
* 对检测到的每个人脸，使用recognize\_face进行识别。
* 在视频帧上标记识别结果。

#### 结束操作

确保在脚本结束时释放摄像头资源并关闭所有窗口，以避免资源泄露。

-------------------------------------------------------------------------------

***Python***

cap.release()

cv2.destroyAllWindows()

-------------------------------------------------------------------------------