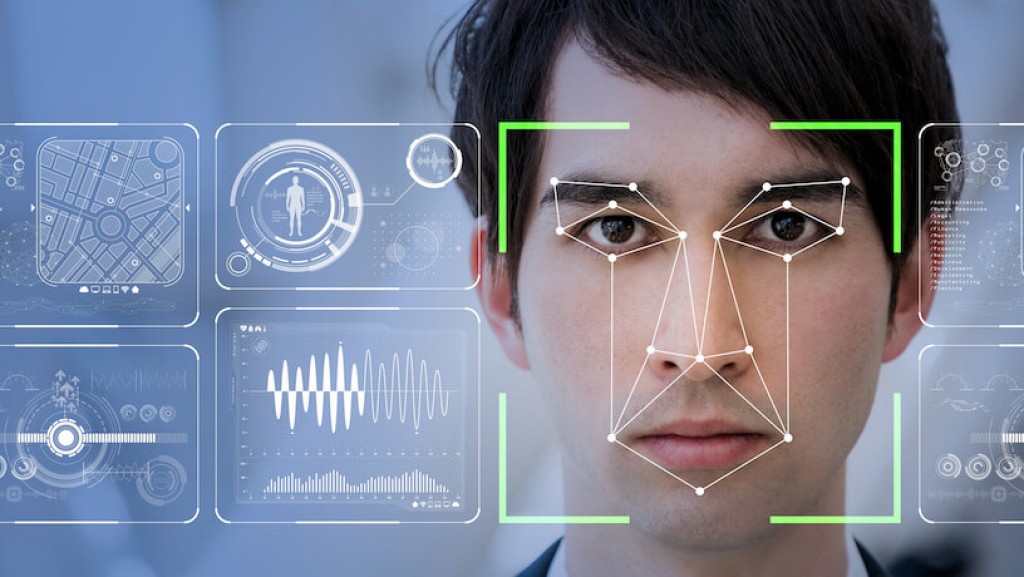
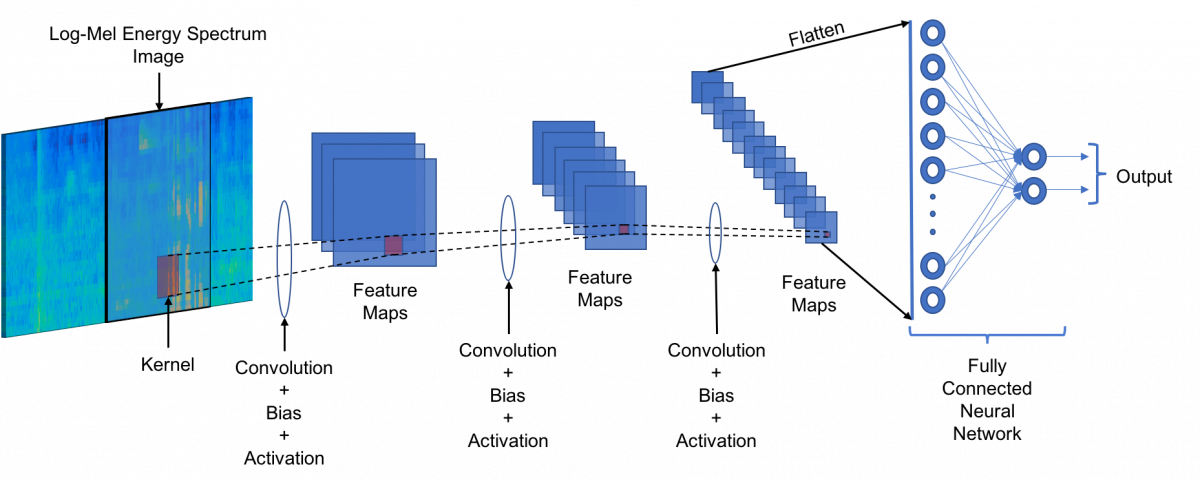
一、简介



本项目采用了独立自主训练的面部识别模型，能够个性化的根据用户的面部特征抽取出有效信息进行高效准确的面部识别。对于模型部分，我们使用了Tensorflow的深度学习框架，使用FaceNet模型进行封装，最终得到我们的个性化专注度识别模型。

二、模型结构

本项目采用了FaceNet结构进行脸部的专注度识别，使用one-shot learning技术对用户存储的图片进行高效提取信息用于之后的人脸比对，使用目标分割算法进行人脸的面部跟踪和视线跟踪，能够准确的追踪用户眼睛的关注点，最后还使用表情识别，对用户的表情进行判别，从而综合得到用户的专注度。



上图是我们的模型示意图，其中output即输出部分实际上为多个输出，分别为表情识别，眼睛视线识别以及人脸比对的01判别。

三、FaceNet介绍

近年来，人脸识别技术取得了飞速的进展，但是人脸验证和识别在自然条件中应用仍然存在困难。本文中，作者开发了一个新的人脸识别系统：FaceNet，可以直接将人脸图像映射到欧几里得空间，空间距离的长度代表了人脸图像的相似性。只要该映射空间生成，人脸识别，验证和聚类等任务就可以轻松完成。文章的方法是基于深度卷积神经网络。FaceNet在LFW数据集上，准确率为0.9963，在YouTube Faces DB数据集上，准确率为0.9512。

四、人脸比对功能

本项目在使用了FaceNet的同时，应用了One-Shoting learning算法，即模型只需要用户的一张照片，就可以高精度的判别下一张图片是不是该用户。这种做法非常符合项目的需要，用户只需要上传一张照片而不是多张照片，从而减轻了他们在注册过程中的负担。它的输入是两个人脸特征，通过人脸比对获得两个人脸特征的相似度，通过与预设的阈值比较来验证这两个人脸特征是否属于同一人（即相似度大于阈值，为同一人；小于阈值为不同）。这个功能用来验证听课的用户是否是该用户本人。

五、用户视线提取功能

视线，又称为注视点估计，是指根据眼睛的位置头部姿态，确定你看的位置，并通过一定的算法估算出来，常用应用场景包括手机的注视解锁，判断广告的投放情况等。常用的方法主要分为三类，一类是基于appearnce的，一类是基于model的，一类是基于feature的。基于apperance的方法，主要是通过眼睛的图像或者通过脸部的图像进行回归，从而获得视线方向或者注释点方向；基于model的方法，主要通过2D反投影回3D计算视线方向，根据眼睛位置计算和屏幕的交点；基于featur的方法，主要通过提取瞳孔特征点，眼睛的眼角等特征，通过FaceNet接Fully Connection Layers方法，获得视线方向。由于用户只有在专注于电脑屏幕时，其视线才会在某个区域内。因为利用用户的视线角度便可以进行有效高准确率的判别。

六、用户表情识别功能

表情特征提取主要采用数学方法，依靠计算机技术对人脸表情的数字图像进行数据的组织和处理，提取表情特征，去除非表情噪声的方法。在某些情况下，特征提取算法提取了图像的主要特征，客观上降低了图像的维数，因此这些特征提取算法也具有降维的作用。

人脸表情的产生是一个很复杂的过程，如果不考虑心理和环境因素，呈现在观察者面前的就是单纯的肌肉运动，以及由此带来的面部形体和纹理的变化。静态图像呈现的是表情发生时单幅图像的表情状态，动态图像呈现的是表情在多幅图像之间的运动过程。因此根据表情发生时的状态和处理对象来区分，表情特征提取算法大体分为基于静态图像的特征提取方法和基于动态图像的特征提取方法。其中基于静态图像的特征提取算法可分为整体法和局部法，基于动态图像的特征提取算法又分为光流法、模型法和几何法。此处我们采用了基于FaceNet的特征提取方法，当用户的表情过于夸张时，我们则认为该用户没有专注听课或对其给予一定的负面评价。

1. 后端对应接口

1. detect\_face(face\_img)

描述：

该方法首先检测给定的图片数据face\_img中是否存在人脸，如果有，则分析用户的专注程度；如果没有，则认为该用户不专心。

参数：

face\_img： base64 编码的二进制图片数据。

返回值:

值为0-1的数字，数字越大表示越专注

调用示例：

with open(“example.png”, 'rb') as img:

            face\_img = base64.b64encode(img.read())

            con=FaceNet().detect\_face(face\_img)

2. create\_face(img, user\_id)

描述：

存储id为user\_id的用户的照片，该照片img是用户的真实照片，可以用来进行后续的判断。

参数:

img：用户的照片数据流

user\_id：用户的ID

返回值：

无

调用示例：

with open(“example.png”, 'rb') as img:

            FaceNet().create\_face(img.read(),”111111”)

3. compare\_face(account, face\_img)

描述：

将照片face\_img与ID为account的用户的真实照片进行比较，判断是否为同一个人。

参数：

account：用户id

face\_img：base64编码的二进制待测照片数据

返回值：

true（是同一个用户）或者false（不是同一个用户）

调用示例：

with open(“test.png”, 'rb') as img:

           face\_img = base64.b64encode(img.read())

           con=FaceNet().compare\_face(face\_img)

八、图片要求

图片格式：JPG(JPEG)，PNG

图片像素尺寸：最小 48\*48 像素，最大 4096\*4096 像素

图片文件大小：2 MB

最小人脸像素尺寸： 系统能够检测到的人脸框为一个正方形，正方形边长的最小值为图像短边长度的 48 分之一，最小值不低于 48 像素。 例如图片为 4096\*3200 像素，则最小人脸像素尺寸为 66\*66 像素。