

# zAI 的人脸识别服务器用法

## 什么是响应式？

响应式在网络通讯过程中是一种工作机制：当我们发送完成一个请求后，必定会收到一条反馈。我们常用的 http 协议，就是一种响应式。

zAI\_Reponse\_FaceServer: 响应式人脸识别服务器

zAI\_Reponse\_FaceClient: 响应式人脸识别客户端

zAI\_Reponse\_FaceClient 提供了 2 个核心功能，保存学习人脸，识别人脸，它们只会在每张照片中央学习+识别人脸。

## 学习人脸的原型方法

**procedure SaveFace(face\_label: SystemString; Scale4x: Boolean; input: TMemoryRaster);**

- **Face\_label**, 人脸的标签，如果同一个人有多张人脸照片，标签就命名同一个人即可
- **Scale4X**, 因为 zAI 的人脸识别都按 720p 标准建模，部分尺寸过小的照片会发生检测不到人脸的情况，这时候，scale4x 是指将小尺寸照片放大 4 倍来寻找照片最中央位置的人脸来学习
- **Input**, 照片光栅数据

使用 SaveFace 方法应该是一次一批的向服务器输入，最好避免间断性的输入。

当 SaveFace 被调用后，空闲 30 秒，zAI\_Reponse\_FaceServer 会在后台自动化的学习人脸。它会自动化生成度量网络，LearnEngine 的 KDTree 向量结构，我们不需要操心服务器的工作，当然，我们可以优化服务器，因为这是随 Demo 发布出来的标准人脸 CS 识别模型。

# 识别人脸的原型方法

**procedure RecFace\_P(input: TMemoryRaster; depthRec: Boolean; OnRecFace: TOnRecFaceP);**

- **Input**, 照片光栅数据
- **depthRec**, 如果在照片中央地区检测不到人脸, 服务器会尝试把人脸放大 4 倍再次检测, 这个参数一般用于处理尺寸很小的人脸照片
- **OnRecFace**, 服务器识别人脸后的返回事件, 在事件中, 至少会包含一个由服务器生成好的识别结果光栅, 以及识别后人脸的框体+标注, 多数时, 我们需要使用服务器的反馈数据来判断人脸的准确度, 诸如: 范冰冰的相似性如果是 90%就是符合, 低于 90%判定不符合。

**RecFace\_P** 可以高频率调用, 但是它并不是针对实时人脸识别而设计, 因为它每次识别都会产生一次识别后的响应动作, 无论成功和识别, 都会给你返回。

上述两个组件是以 Demo 形式存在于 Demo\Reponse\_FaceServer 目录中的, 运行它建议使用需要 GPU+cuda。

上述 Demo 包含了: 光栅处理, 通讯, face 检测, face 对齐, face 度量化, face 向量匹配若干技术点。

经过优化的 face 后台可以承受数千次/秒的人脸识别负载。

By.qq600585

2019-4