

Mini人脸识别接口

修改历史

Version	Contributor	Date	Change Log
v1.0	吴仕意	18/08/09	新增Mini人脸识别接口说明

SDK

- 实体类结构

```
public class FacelInfo implements Parcelable {  
    /**  
     * 录入的人脸Id, 为空时表示为陌生人  
     */  
    private String id;  
    /**  
     * 录入的人脸名字  
     */  
    private String name;  
    /**  
     * 录入的头像  
     */  
    private String avatar;
```

```
/**
 * 录入的时间
 */
private long time;
/**
 * 角度 pitch 上下偏移:[-30,30]
 */
private float pitch = 0f; //角度 pitch 上下偏移:[-30,30]
/**
 * 角度 pitch 上下偏移:[-30,30]
 */
private float yaw= 0f; //角度 yaw 左右偏移:[-30,30]
/**
 * 角度 roll:平面旋转[-180,180]
 */
private float roll= 0f; //角度 roll:平面旋转[-180,180]
/**
 * 角度 yaw 左右偏移:[-30,30]
 */
private boolean famous;
/**
 * 性别 (女)0~100(男) (仅限人脸分析接口)
 */
private int gender; //性别 (女)0~100(男) (仅限人脸分析接口)
/**
 * 年龄 (仅限人脸分析接口)
 */
private int age; //年龄 (仅限人脸分析接口)
/**
 * 人脸框左上角x坐标, 坐标原点为图片左上角
```

```
*  
* @return int  
*/  
private int x;  
  
/**  
* 人脸框左上角y坐标,坐标原点为图片左上角  
*  
* @return int  
*/  
private int y;  
/**  
* 人脸的宽度  
*  
*/  
private int width;  
/**  
* 人脸的高度  
*  
*/  
private int height;  
/**  
* 预览图片的宽度  
*  
*/  
private int imgWidth;  
/**  
* 预览图片的高度  
*  
*/  
private int imgHeight;  
  
/**  
* 是否陌生人  
*  
* @return is stranger  
*/
```

```
public boolean isStranger() {  
    return TextUtils.isEmpty(id);  
}  
}
```

- java 接口

```
/**  
 * 检测机器人当前是否处于录入过程  
 *  
 * @param responseListener 结果回调  
 */  
public void checkFaceInsertState(final ResponseListener<Void> responseListener);  
  
/**  
 * 人脸注册skill 启动接口  
 *  
 * @param userId 如果是客户端触发带上当前用户id, 以便调用退出人脸录入的会话验证  
 * @param name 录入人的名称  
 * @param responseListener 结果回调,回调结果带上userId  
 */  
public void apiFaceRegister(final String userId,final String name, final ResponseListener<String> responseListener) ;  
  
/**  
 * 退出人脸录入  
 *  
 * @param responseListener 结果回调  
 */  
public void apiStopRegister(String userId,final ResponseListener<Void> responseListener);  
  
/**  
 * 更新人脸信息  
 *  
 * @param id 要更新的人脸id  
 * @param name 要更新的名称
```

```
* @param responseListener 结果回调
*/
public void update(final String id, final String name, final ResponseListener<Void> responseListener);

/**
 * 删除人脸信息
 *
 * @param ids 要删除的id列表
 * @param responseListener 结果回调
 */
public void delete(final ArrayList<String> ids, final ResponseListener<Void> responseListener)

/**
 * 查询当前已录入的人脸信息
 *
 * @param page 查询页码数
 * @param pageSize 每页个数
 * @param responseListener 结果回调
 */
public void query(final int page, final int pageSize, final ResponseListener<ArrayList<FaceInfo>> responseListener)

/**
 * 即时人脸离线检测,检测当前人脸, 检测到人脸立即返回结束
 *
 * @param timeout 未检测到人脸的超时时间,单位是秒, 最小时间是7秒小于7的值当7处理
 * @param responseListener 结果回调
 */
public void faceDetect(long timeout, final ResponseListener<List<FaceInfo>> responseListener)

/**
 * 人脸分析, 检测当前画面的人脸, 并进行分析, 获得年龄和性别的等信息, 注: 不包括是否熟人的信息
 *
 * @param timeout 未检测到人脸的超时时间,单位是秒, 最小时间是7秒小于7的值当7处理
 * @param responseListener 结果回调
 */
public void faceAnalyze(long timeout, final ResponseListener<List<FaceInfo>> responseListener)
```

```
/**
 * 人脸识别，检测当前画面的人脸信息（是否熟人，不包括年龄、性别等信息）。
 *
 * @param timeout      人脸信息
 * @param responseListener 结果回调
 */
public void faceRecognize(long timeout, final ResponseListener<List<FacelInfo>> responseListener)

/**
 * 移动舵机自动寻找人脸，返回寻找到的人脸
 *
 * @param responseListener 结果回调
 */
public void findFace(long timeout, final FaceFindListener responseListener)

/**
 * 暂停找人脸
 *
 * @param responseListener 结果回调
 */
public void pauseFindFace(final ResponseListener<Void> responseListener)

/**
 * 退出找人脸
 *
 * @param responseListener 结果回调
 */
public void stopFindFace(final ResponseListener<Void> responseListener)

/**
 * 退出人脸追踪识别
 *
 * @param responseListener 结果回调
 */
public void apiStopFaceTrack(final ResponseListener<Void> responseListener)

public interface FaceTrackListener {
```

```
/**
 * 开始.
 */
public void onStart();
/**
 * 人脸数量发生变化.
 */
public void onFaceChange(List<FaceInfo> faceInfos);
/**
 * 停止.
 */
public void onStop();
/**
 * 开启失败.
 */
public void onFail(int code, String msg);

}

public interface FaceFindListener extends FaceTrackListener {
/**
 * 暂停回调.
 */
public void onPause();

}
```