百度AI活体检测JS-SDK

方案介绍

- 静默活体：实时检测人脸质量，包含人脸检测，屏幕距离检测，环境光检测，人脸空间角检测，人脸超时检测
- 动作活体：静默活体 + 动作检测（张嘴，眨眼，向左转头，向右转头，向上抬头，向下低头）
- 炫瞳活体：静默活体 + 随机张嘴或者眨眼动作 + RGB颜色序列闪光

交付包清单

1. demos:包含Vue版简单demo实现。
2. models:引入至项目的模型文件，放置服务端静态资源目录下即可，具体参见「引入模型文件」。
3. SDK:活体检测需要调用的sdk包，具体参见「引入sdk」。

使用说明

一：引入sdk

```
# 模块化引入
import BDFaceSdk from './BDFaceSdk.esm.min.js';
```

二：引入模型文件

将models目录放置服务端静态资源目录下即可。以demo为例，使用vue-cli启动本地开发服务，所以将models目录放置于public目录下，最终上线需要保证模型文件能通过url正确访问到。

三：调用SDK

1. 初始化sdk:

在页面加载完成后进行初始化操作，以demo为例，在mounted阶段进行初始化init操作，并调用加载模型方法loadModels。

```
import BDFaceSdk from './BDFaceSdk.min.js';

const demo = new BDFaceSdk({
    // 对应html中id为video的dom元素id
    video: 'face-live-detection-video',
    // 活体方案类型：  0-静默检测；1-动作检测；2-炫瞳检测
    solutionType: 1,
    // 如果是动作检测或炫瞳检测，自定义需要做的动作序列：
    // 0-眨眼；1-张嘴；2-向右转头；3-向左转头；4-向上抬头；5-向下低头
    code: '135',
    // 是否随机动作，开启后会随机指定一个动作，如果不是随机的可以通过code属性进行灵活配置
    randomMotionDetection: false,
    // 是否开启录制功能
    needToRecord: true,
```

```
    // 权限约束
    constraints: {
        video: {
            width: {
                min: 320,
                ideal: 320,
                max: 320
            },
            height: {
                min: 240,
                ideal: 240,
                max: 240
            }
        },
        audio: false
    },
    // 模型文件url前缀，默认是当前页面根路径,可不传
    modelPath: 'https://xxx.xxx.com',
    // 可通过调节遮挡部位阈值，来调整遮挡判断。
    occlusion: {
        eye: 0.8,
        mouse: 0.8,
        nose: 0.8,
        leftFace: 0.8,
        rightFace: 0.8,
        chin: 0.8
    }
});
```

参数说明：

参数	类型	说明	是否为必填
video	string	页面中镶嵌video元素的id	是
solutionType	0 | 1 | 2	方案类型（0静默，1动作，2炫瞳）	是
code	string	动作序列（方案类型为动作时必传）	否
modelPath	string	模型文件url前缀，默认为根路径（需要自定义路径才传）	否
needToRecord	boolean	是否需要对实时检测全流程进行视频录制，默认是 false	否
timeout	number	检测超时时间，默认是15000毫秒	否
actionTimeout	number	动作检测超时时间，默认5000毫秒	
minFaceRatio	number	最小检测人脸阈值，默认是0.2	否
maxFaceRatio	number	最大检测人脸阈值，默认是0.8	否
constraints	{video: boolean; audio: boolean}	权限约束，如果需要同时开启相机和音频权限，请设置成{video: true, audio: true}	否
randomMotionDetection	boolean	在炫瞳前的动作检测是否是随机的，如果不是随机的可以通过code属性进行灵活配置	否
maxSizeKB	number	最大压缩产出图片的大小，以KB为单位，默认是70	否
authUrl	string	鉴权服务地址	否
occlusion	boolean | {eye:number; mouse:number;nose:number; leftFace:number; rightFace:number; chin:number;}	// 默认值:// 遮挡阈值occlusion: {// 遮挡眼睛eye: 0.8,// 遮挡嘴巴mouse: 0.8,// 遮挡鼻子nose: 0.8,// 遮挡左脸leftFace: 0.8,// 遮挡右脸rightFace: 0.8,// 遮挡下巴chin: 0.8}// 不使用遮挡能力 - 传false	否

code 说明

值	对应动作
0	眨眼
1	张嘴
2	向右转头
3	向左转头
4	向上抬头
5	向下低头

四、绑定事件回调

在验证过程中会抛出一些回调事件，用户可以监听这些事件，自行做一些处理，比如loading，进度条，页面打光。以demo为例，在初始化阶段调用bindEvents对这些事件进行绑定。

1. status事件

```
demo.on('status', status => {
    // 根据status做一些处理
});
```

当前包含以下status：

状态值	说明
loading	初始化loading中
loading_over	初始化loading完成

2. tip事件

```
demo.on('tip', ({code, text}) => {
    // 根据code或者text在页面中给出提示文案
});
```

当前包含以下code和text：

code	text	说明
empty	未检测到人脸	没有检测到人脸
multiple	请保持单人入镜	检测到多张人脸
outside	请将人脸移入框内	人脸不在屏幕框内
close	请远离屏幕	人脸距离屏幕太近
far	请靠近屏幕	人脸距离屏幕太远
light	请调暗环境光线	环境光太亮
dark	请调亮环境光线	环境光太暗
rotate	请保持正脸	人脸偏转角度过大
blink	眨眨眼	提醒用户眨眼
mouthopen	张张嘴	提醒用户张嘴
headright	向右转头	提醒用户向右转头
headleft	向左转头	提醒用户向左转头
lookup	向上抬头	提醒用户向上抬头
lookdown	向下低头	提醒用户向下低头
action_over	动作核验完成	动作核验完成
color_ready	屏幕即将闪烁，请保持正脸	准备开始炫瞳流程
color_running	变光中，请保持正脸	炫瞳流程进行中
ok	请保持不动	当前人脸符合要求
timeout	采集超时	采集超时（15s未检测到人脸或未完成动作）
actionTimeout	动作采集超时	动作采集超时（5s）

3. action事件

```
demo.on('action', index => {
    // 返回已经完成的动作数量，用户每次做完一个动作，index就会+1
    // index范围 [1, code.length]
});
```

4. color事件

```
demo.on('color', (currentColorIdx, currentColorValue) => {
    // 返回当前正在打光的颜色索引和颜色值
    // currentColorIdx范围 [0, 2]
    // currentColorValue范围 ['#db357f', '#1eb848', '#371dc0']
    // 用户可以根据currentColorIdx和currentColorValue自行打光，具体实现可参考示例demo
   });
```

5. success事件

```
 demo.on('success', ({images, video}) => {
     // images：静默活体、动作活体、炫瞳活体的四张质量检测最优图片
     // video：录制视频blob文件
 });
```

6. fail事件

```
demo.on('fail', params => {
    // params: [type, images]
    // params[0]：失败类型
    // params[1]: 质量检测图片
    const type = params[0];
});
```

当前包含以下type:

Type	说明
unsupported	浏览器不支持webrtc
reject	浏览器没有摄像头权限或用户拒绝授权
timeout	流程超时（需补充至逻辑内）
pdFail	炫瞳检测失败
action_timeout	检测超时（需补充至逻辑内）
notallowed	相机权限被拒绝

五、方法：

1. 开始验证

```
// 开始验证，此时会唤起摄像头。并在id为videoWrapper的标签内插入video标签。
// 注：该方法只会在首次验证的时候调用，后续需要用restart方法
demo.start();
```

2. 重新验证

```
// 当动作超时或者采集流程超时，需要重新发起认证，这时候用restart方法
// 与start方法的不同之处在于不会重新执行插入video和一些初始化操作
demo.restart();
```

3. 销毁流程

```
// 销毁验证流程，建议在组件卸载时调用
demo.destroy();
```

六：安全版本加密解密代码示例

```
// 代码示例：
// 检测成功，回调函数抛出加密后图片
this.bdFaceLiveDetection.on('success', result => {
    this.tip = '检测完成';
    const params = {
        face: result.face,
        jr: result.jr
    };
    this.fetchFaceSec(params);
});

/**
 * 人脸图片解密与风险分析接口
 * 注意：由于ak sk 较敏感，客户方调用该解密需要在服务端进行，当前使用的是默认的ak，sk，
 *       实际调用时使用客户自己的ak，sk
 */
 fetchFaceSec(params) {
    const timestamp = Date.now();
    const appkey = 'xxx';
    const sign = md5(appkey + timestamp + 'xxx');
    fetch(`/face-sec/v3/decode?sign=${sign}&timestamp=${timestamp}&appkey=${appkey}`, {
        method: 'POST',
        body: JSON.stringify({
            encryType: '0',
            app: 'universe',
            data: JSON.stringify(params)
        }),
        headers: new Headers({
            'Content-Type': 'application/json'
        })
    })
    .then(response => response.json())
    .then(res => {
        // ...
        this.fetchFaceLiveness([res.result.face.images[0]]);
```

```
    });
},
/**
 * 图片活体检测结果接口
 */
fetchFaceLiveness(images) {
    const data = images.map(item => {
        return {
            image: item.substring(item.indexOf(',') + 1),
            image_type: 'BASE64',
            face_field: 'spoofing'
        };
    });
    fetch('/face-api/v3/face/liveness?appid=1', {
        method: 'POST',
        body: JSON.stringify(data),
        headers: new Headers({
            'Content-Type': 'application/json'
        })
    })
    .then(response => response.json())
    .then(res => {
        this.bdFaceLiveDetection.destroy();
        if (res.error_code === 0) {
            // 成功
            // ..
        }
        else {
            // 失败
            // ...
        }
    });
},
```

客户常见问题

1. 提示「浏览器不支持」：

- 先让客户检查下部署的是否为https协议
- 然后检查当前使用的浏览器是否支持webrtc，目前遇到的不支持的环境：企业微信ios内置浏览器、钉钉安卓内置浏览器、夸克浏览器

2. 提示「获取相机权限失败」：

- 检查客户是否开启相机授权

浏览器兼容性

浏览器	最低支持版本	是否支持
Chrome	53	是
Edge	12	是
Firefox	68	是
Opera	40	是
Safari	11	是
IE	--	否
360极速浏览器	--	否