SLSkinAnalysisQCloud V2.0.0 SDK iOS平台接入指南

`SLSkinAnalysisQCloud` 是 `HETSkinAnalysisSDK` 的升级版本，改进架构并优化了拍照性能和体验，内部集成了 `QCloud` 云服务`。

变更记录说明

日期	版本号	作者	描述
2020年06月15日	1.0.6	马远征	1、增加了鱼尾纹维度；2、增加了算法超时错误码
2020年06月15日	1.0.7	李孝宇	1、增加了遮挡物字段（faceShelter）；2、增加了测肤记录ID(photographId)
2020年07月23日	1.0.8	李孝宇	1、增加了眉间纹；2、细纹 Wrinkles 下增加面积字段
2020年08月20日	1.0.9	李孝宇	1、增加了眉间纹；2、细纹 Wrinkles 下增加面积字段
2020年09月24日	1.1.0	马远征	1、增加文案、icon、蒙版、可视化图片输出

日期	版本号	作者	描述
2020年10月16日	2.0.0	马远征	1.新增油分整脸情况字段oilOverall；2.新增水分整脸情况字段moistureOverall；3.新增敏感整脸情况字段sensitivityOverall；4.新增对应原图的人脸区域坐标字段orgImageFaceLocation。5、新增遮挡物检测功能

一、平台授权

`SDK` 需要配置相应的 `AppId` 和 `AppSecret` ，请联系相关客服人员获取。

二、快速集成

SDK支持 iOS 9.0 以上设备，请保持Xcode开发工具升级到最新版本。

1、资源下载

> ✅ 肤质分析 SDK 下载 SLSkinAnalysisQCloud.framework
> ✅ Demo下载 SLSkinAnalysisDemo

2、Xcode 集成

1、将`SLSkinAnalysisQCloud.framework` 拖到你的项目中，将 `Embed` 设置为 `Embed & Sign`

2、添加必要的链接库文件，并在 `Other Link Flags` 选项添加 `-ObjC`

- libc++.tdb

3、在需要使用的地方导入头文件 `#import <SLSkinAnalysisQCloud/SLSkinAnalysisQCloud.h>` 即可进入相关开发

三、接入指南

在调用相关接口前请先下载`SDK`和`Demo`，熟悉接口和调用逻辑。

3.1、注册 AppId 和 AppSecret

```
// 优先导入头文件
#import <SLSkinAnalysisQCloud/SLSkinAnalysisQCloud.h>

// 注册
SLSARegister(@"31298", @"145a2540f00147e89dc5e33b6842f74c");
```

3.2 相机调用

在项目工程 `Info.plist` 中添加 `Privacy - Camera Usage Description` 隐私描述 如： `App拍照测肤需要调用您的相机`

3.2.1 访问授权

在调用相机拍摄前需要优先判断是否拥有相机访问许可，如果未授权需要请求授权访问。

```
AVAuthorizationStatus status = [AVCaptureDevice authorizationStatusForMed
iaType:AVMediaTypeVideo];
if (status == AVAuthorizationStatusAuthorized) {
    // 初始化相机
    return;
}
@weakify(self);
[AVCaptureDevice requestAccessForMediaType:AVMediaTypeVideo completionHan
dler:^(BOOL granted) {
    SLSAAsyncMain(^{
        if (granted) {
            // 初始化相机
            return;
        }
        // 无相机访问许可，做对应的处理
        [self showAlert:@"无相机访问许可，请更改隐私设置允许访问相机" doneHandle
r:^{
            @strongify(self);
            [self.navigationController popViewControllerAnimated:YES];
        }];
    });
}];
```

3.2.2 相机初始化

拥有相机访问权限后，可按如下方式初始化相机

```
NSError *error;
_camera = [[SLSACamera alloc]init];
_camera.delegate = self;
BOOL ret = [_camera prepareCamera:AVCaptureDevicePositionFront error:&err
or];
if (!ret) {
    // 相机初始化失败，做失败处理
    NSLog(@"---相机设备初始化失败--%@",error);
    @weakify(self);
    [self showAlert:@"相机初始化出错" doneHandler:^{
        @strongify(self);
        [self.navigationController popViewControllerAnimated:YES];
    }];
    return;
}
/// 调用处理，如插入layer图层并启动相机视频帧采集
if (_camera.prepared) {
    _camera.videoPreviewLayer.frame = self.view.bounds;
}
[self.view.layer insertSublayer:_camera.videoPreviewLayer atIndex:0];
[_camera startRunning];
```

3.2.3 videoPreviewLayer 设置

`AVCaptureVideoPreviewLayer` 提供了相机画面的异步渲染，如果不需要自行渲染相机数据，可以调用它。

- 将相机预览图层插入视图图层索引0位置

```
[self.view.layer insertSublayer:_camera.videoPreviewLayer atIndex:0];
_camera.videoPreviewLayer.frame = self.view.bounds;
```

- 调整预览图层的 frame

```
- (void)viewWillLayoutSubviews {
    [super viewWillLayoutSubviews];
    if (self.camera && self.camera.prepared) {
        self.camera.videoPreviewLayer.frame = self.view.bounds;
    }
}

- (void)viewDidLayoutSubviews {
    [super viewDidLayoutSubviews];
    if (self.camera && self.camera.prepared) {
        self.camera.videoPreviewLayer.frame = self.view.bounds;
    }
}
```

3.2.4 运行控制

- 判断相机是否正在采集视频帧

```
BOOL ret = [self.camera isRunning];
if (ret) {
    /// 相机正在运行
}
```

- 启动相机视频帧录制

```
/// 异步启动相机
[self.camera startRunning];
/// 同步启动相机
[self.camera startRunning:NO];
```

- 停止视频帧录制

```
/// 异步停止相机拍摄
[self.camera stopRunning];
```

3.2.5 摄像头切换

- 判断是否是后置摄像头

```
/// 异步停止相机拍摄
BOOL ret = [self.camera isCameraPositionBack];
```

- 获取摄像头位置

```
AVCaptureDevicePosition position = [self.camera getCameraPosition];
```

- 异步切换摄像头

```
/// 相机正在拍照，不执行切换
if ([self.camera isCapturingStillImage]) {
    return;
}
// 获取当前摄像头方向并设置相反的切换方向
AVCaptureDevicePosition devicePosition = [self.camera getCameraPosition];
if (devicePosition == AVCaptureDevicePositionBack) {
    devicePosition = AVCaptureDevicePositionFront;
}
else {
```

```
        devicePosition = AVCaptureDevicePositionBack;
    }
    /// 切换摄像头
    @weakify(self);
    [self.camera setCameraPosition:devicePosition result:^(AVCaptureDevicePos
    ition position) {
        @strongify(self);
        /// 得到摄像头切换的方向 position 做对应的处理
        BOOL isFrontCamera = [self.camera isCameraPositionBack];
        self.voiceTextLabel.text = isFrontCamera ? @"请平视前置摄像头" : @"请平视
    后置摄像头";
        self.switchButton.selected = (position ==
    AVCaptureDevicePositionBack);
    }];
```

3.2.6 静态图片采集

调用如下接口进行静态图片拍摄

```
    /// 正在拍照
    if ([self.camera isCapturingStillImage]) {
        return;
    }
    @weakify(self);
    [self.camera takePhotosAsynchronously:^(CMSampleBufferRef  _Nonnull image
    DataSampleBuffer, NSError * _Nonnull error) {
        /// 拍照结果buffer回调
        @strongify(self);
        SLSAAsyncMain(^{

        });
    } result:^(NSData * _Nonnull imageData, NSError * _Nonnull error) {
        if (error) {
        /// 拍照错误
            return;
        }
        /// 得到原始图像进行处理
        UIImage *originImage = [[UIImage alloc]initWithData:imageData];
    }];
```

3.2.7 清理资源

使用完毕释放资源

```
    - (void)dealloc {
        if (_camera) {
            [_camera clear];
            _camera = nil;
```

```
    }
}
```

3.3 buffer 处理

`SLSAVideoBufferAnalysisEngine`分析引擎用于辅助拍摄符合要求的清晰人脸图像，通过此引擎可以控制拍摄距离、光亮、人脸在屏幕位置、识别人脸遮挡物等。

3.3.1 configure

`SLSAVideoBufferAnalysisConfiguration`允许你设置buffer分析阈值，如：光亮、距离、分析选项、稳定帧等

```
SLSAVideoBufferAnalysisConfiguration *config = [[SLSAVideoBufferAnalysisC
onfiguration alloc]init];
config.minDistance = 0.55; // 最小相对距离
config.maxDistance = 0.95; // 最大相对距离
config.minYUVLight = 60; // 最小亮度
config.maxYUVLight = 220; // 最大亮度
config.options = SSVideoBufferAnalysisAll;// 检测所有选项
config.minStableFramesToCaptureStillImage = 3; // 最小拍照稳定帧，3次满足拍照
要求输出拍照状态
config.minStableFramesToOutputState = 3; // 静音模式最小稳定帧，3此满足输出当前
状态
```

3.3.2 自定义语音

`buffer`分析引擎提供了默认的语音`SLSADefaultVoiceConfiguration`,如果需要自定义声音，实现`SLSAVoiceConfigDelegate`协议即可

```
@interface SLSAMyCustomVoiceConfiguration : NSObject <SLSAVoiceConfigDele
gate>

@end

@implementation SLSAMyCustomVoiceConfiguration

#pragma mark - Delegate

- (nullable SLSAVoiceItem*)getVoiceItemByVideoBufferAnalysisState:(SSVide
oBufferAnalysisState)state frontCamera:(BOOL)isFrontCamera {
    /// 实现非遮挡物状态语音
}
- (nullable SLSAVoiceItem*)getVoiceItemByDetectedFaceShelters:(NSArray<SL
SAFaceShelterItem*>*)shelters {
    /// 实现遮挡物状态语音
```

```
}
@end
```

3.3.3 人脸状态检测

- 初始化 buffer 分析引擎

```
SLSAVideoBufferAnalysisConfiguration *config = [[SLSAVideoBufferAnalysisC
onfiguration alloc]init];
config.minDistance = 0.55;
config.maxDistance = 0.95;
SLSAMyCustomVoiceConfiguration *voiceConfig = [[SLSAMyCustomVoiceConfigur
ation alloc]init];
_bufferAnalysisEngine = [[SLSAVideoBufferAnalysisEngine alloc]initWithCon
figuation:config voiceConfig:voiceConfig];
```

- buffer分析

```
- (void)captureOutput:(AVCaptureOutput *)output didOutputSampleBuffer:(CM
SampleBufferRef)sampleBuffer faceObjects:(NSArray *)faceObjects fromConne
ction:(AVCaptureConnection *)connection {
    AVCaptureDevicePosition position = [self.camera getCameraPosition];
    [self.bufferAnalysisEngine analysisVideoBuffer:sampleBuffer position:
position faces:faceInfoArray renderRect:self.renderRect
boundingRect:self.renderRect targetFace:^(CGRect faceRect) {
        NSLog(@"--人脸框--%@",NSStringFromCGRect(faceRect));
        SLSAAsyncMain(^{
            @strongify(self);
            [self.faceRectDraw drawFaceRect:faceRect];
        });
    } result:^(SLSAVideoBufferAnalysisResult * _Nonnull result) {
        NSLog(@"--result--%@",result);
        @strongify(self);
        /// 处理人脸状态
        /// code
        if (result.state == SSVideoBufferAnalysisStateWillTakePhoto) {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self performSelector:@selector(prepareToTakePhoto) withO
bject:nil afterDelay:3.0];
            });
        }
        else {
            dispatch_async(dispatch_get_main_queue(), ^{
              NSLog(@"取消拍照");
             [NSObject cancelPreviousPerformRequestsWithTarget:self selec
tor:@selector(prepareToTakePhoto) object:nil];
            });
        }
    }];
```

```
}
```

3.4 静态图片检测

可使用静态图片检测接口识别图片是否符合大数据肤质分析要求，如果不做处理且不满足要求，后台算法服务器会返回对应的错误

```
/// 你的检测图像
UIImage *originImage = [[UIImage alloc]initWithData:imageData];

/// 检测拍摄的静态图片的可用性
SSStillImageAnalysisOptions options = SSStillImageAnalysisNone;
options = (options | SSStillImageAnalysisFaceFeature);
options = (options | SSStillImageAnalysisFaceShelters);
options = (options | SSStillImageAnalysisAspectRedio);
options = (options | SSStillImageAnalysisPixels);
options = (options | SSStillImageAnalysisSize);
options = (options | SSStillImageAnalysisPixels);

SLSAStillImageAnalysisConfiguration *config = [[SLSAStillImageAnalysisCon
figuration alloc]init];
config.options = options;
config.maxPixels = 5000000;
config.maxImageWidth = 2000;
config.maxImageHeight = 2500;

SLSAStillImageAnalysisEngine *engine = [[SLSAStillImageAnalysisEngine all
oc]initWithConfiguration:config];
NSError *stillImageError;
if (![engine isValidStillImage:originImage error:&stillImageError]) {
    [QMUITips showError:stillImageError.localizedDescription];
    return;
}
/// 上传分析
```

3.5 上传与分析

采集到符合要求的正面清晰人脸照片后可调用以下接口进行上传和分析

```
[QMUITips showLoading:@"图片上传中" inView:self.view];
_dataEngine = [[SLSAFaceDataAnalysisEngine alloc]init];
[_dataEngine uploadImage:image progress:^(float progress) {
    NSLog(@"--上传进度--%@",@(progress));
} result:^(NSString * _Nonnull imageURL, NSError * _Nonnull error) {
    [QMUITips hideAllTips];
```

```
    if (error) {
        [QMUITips showError:error.localizedDescription];
        return;
    }
    NSLog(@"---上传成功--%@",imageURL);
    SLSkinAnalysisDecryptQCloudImageURL(imageURL, ^(NSString *decryptedUR
L, NSError *error) {
        if (error) {
            NSLog(@"--图片解密错误--%@",error);
            return;
        }
        NSLog(@"--图片解密成功--%@",decryptedURL);
    });

    /// 肤质分析
    [self.dataEngine analysisWithImageURL:imageURL result:^(NSDictionary
* _Nonnull responseJSON, NSError * _Nonnull error) {
        [QMUITips hideAllTips];
        if (error) {
            NSLog(@"----error----%@",error);
            [QMUITips showError:error.localizedDescription];
            return;
        }
        NSLog(@"----responseJSON----%@",responseJSON);
        [QMUITips showSucceed:@"肤质信息分析成功"];
    }];
}];
```

四、肤质分析功能说明

4.1 能力介绍

- **性别识别**：识别男女；
- **肤色识别**：识别肤色，涵盖亮白、红润、自然、小麦、暗哑等肤色；
- **脸型识别**：识别脸型类别，涵盖圆脸、鹅蛋脸、心形脸、方脸等脸型；
- **黑头识别**：检测黑头个数和严重程度；
- **毛孔检测**：检测毛孔个数和严重程度；
- **痘痘检测**：检测痘痘类别、个数、严重程度、脸部位置，涵盖痘后红斑、凹陷瘢痕、脓包、结节囊肿等类别；
- **眼型识别**：识别眼型类别，涵盖杏眼、丹凤眼、桃花眼等眼型；
- **眉形识别**：识别左右眉形类别，涵盖双燕眉、平直眉、秋波眉等眉形；
- **卧蚕识别**：识别是否有卧蚕；

- **细纹检测**：检测细纹类别和严重程度，涵盖抬头纹、法令纹、泪沟、笑肌断层、鱼尾纹等类别；
- **色素斑检测**：检测色素斑类别、严重程度、脸部位置，涵盖雀斑、黄褐斑、隐藏斑等类别；
- **眼袋检测**：检测是否有眼袋和严重程度；
- **敏感检测**：检测敏感类别和严重程度；
- **油分检测**：检测油分状况，包含油分列表、油分面积、油分面积占比、严重程度；
- **水分检测**：检测水分状况，包含水分列表、水分面积、水分面积占比、严重程度；
- **肤质类型检测**：检测肤质类别；
- **肌龄检测**：检测皮肤年龄；
- **黑眼圈检测**：检测黑眼圈类别和严重程度；
- **脂肪粒检测**：检测缺水类别和严重程度；
- **图片质量检测**：检测图片光照和是否模糊；
- **人脸姿态检测**：检测人脸姿态角度；
- **遮挡物检测**：检测人脸遮挡物（帽子、刘海、眼镜、鼻贴、口罩、面膜）；
- **相似明星脸检测**：检测相似明星脸：明星名称、相似度、明星图片；

4.2 类别描述

属性	类别
性别	男、女
肤色	黝黑、暗哑、小麦、自然、红润、亮白
脸型	方脸、圆脸、鹅蛋脸、心形脸
痘痘	凹陷瘢痕、痘后红斑、粉刺、炎症丘疹、结节囊肿、脓包
眼型	杏眼、小鹿眼、铜铃眼、睡龙眼、丹凤眼、瑞凤眼、睡凤眼、月牙眼、桃花眼、柳叶眼、狐媚眼、孔雀眼
眉形	柳叶眉、平直眉、秋波眉、秋娘眉、双燕眉、水弯眉
细纹	抬头纹、法令纹、泪沟、笑肌断层
色素斑	黄褐斑、雀斑、隐藏斑、黑痣
敏感	耐受、敏感
水分	滋润、敏感性缺水、油脂性缺水、老化性缺水
肤质类型	干性、中性偏干、中性、混合性偏干、混合性、混合性偏油、油性
黑眼圈	正常、血管型、色素型
脂肪粒	栗丘疹

图片质量	光照：正常、黑暗、过曝、光照不均匀；模糊：正常、模糊
人脸姿态	Pitch、 Roll、Yaw
遮挡物	帽子、刘海、眼镜、鼻贴、口罩、面膜

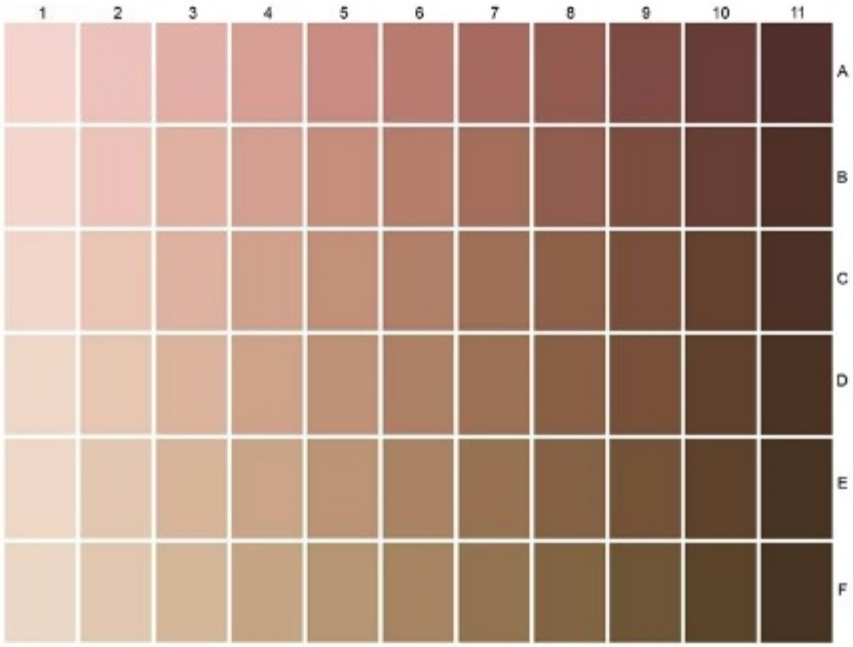
4.3 注意事项

眼型返回值规则：眼型返回值由 eyelid，narrow，updown 三个值组合而成，如返回值 `"eyelid": 2,"narrow": 2,"updown": 1` 代表丹凤眼，具体计算规则见下表。（**注：**下表的组合值为简写，如 `[1、1、1]` 表示 `"eyelid": 1,"narrow": 1,"updown": 1` ）

组合值	代表眼型
[1、1、1]	铜铃眼
[1、1、2]	睡龙眼
[2、1、1]	杏眼
[2、1、2]	小鹿眼
[2、2、1]	丹凤眼
[2、2、2]	睡凤眼
[1、2、1]	瑞凤眼
[1、2、2]	月牙眼
[2、3、1]	桃花眼
[2、3、2]	柳叶眼
[1、3、1]	狐媚眼
[1、3、2]	孔雀眼

肤色返回值规则：肤色返回值由肤色卡行列组合而成，如返回值 F_3 代表第 F 行第 3 列的肤色块，肤色卡见下表。

肤色卡



结果返回

名称	数据类型	说明
blackHead	object	黑头
level	number	严重等级（1-无，2-极少，3-轻度，4-中度，5-重度）
number	number	黑头数量
maskPath	String	黑头图层图片
pore	object	毛孔
level	number	严重等级（1-紧致，2-轻度，3-中度，4-重度）
number	number	毛孔数量
maskPath	String	毛孔图层图片
facecolor	String	肤色（F_3 代表第F行第3列的肤色块)
faceshape	String	脸型（H-心形脸，O-鹅蛋脸, S-方脸, R-圆脸）
acnes	list	痘痘列表
acneTypeId	number	痘痘类型（1-红斑，2-粉刺，3-炎症性丘疹，4-脓包，5-凹陷性瘢痕，6-结节）
level	number	严重等级（1-无，2-轻度，3-中度，4-重度）

名称	数据类型	说明
number	number	痘痘数量
facePart	number	部位（0-脸部无痘痘，1-额头，2-鼻子，3-左脸，4-右脸，5-下颌）
eyeshape	object	眼型
eyelid	number	单双眼皮（1-单眼皮，2-双眼皮）
narrow	number	眼睛宽窄（1-大眼，2-正常眼，3-眯缝眼）
updown	number	外眼角的上扬或下垂（1-上扬，2-下垂）
[1、1、1]		铜铃眼
[1、1、2]		睡龙眼
[2、1、1]		杏眼
[2、1、2]		小鹿眼
[2、2、1]		丹凤眼
[2、2、2]		睡凤眼
[1、2、1]		瑞凤眼
[1、2、2]		月牙眼
[2、3、1]		桃花眼
[2、3、2]		柳叶眼
[1、3、1]		狐媚眼
[1、3、2]		孔雀眼
eyebrow	object	眉形
left	number	左眼眉形（1、柳叶眉，2-平直眉，3-秋波眉，4-秋娘眉，5-双燕眉，6-水弯眉）
right	number	右眼眉形（1、柳叶眉，2-平直眉，3-秋波眉，4-秋娘眉，5-双燕眉，6-水弯眉）
pouch	object	眼袋
exist	number	有无眼袋（1-无眼袋，2-有眼袋）
level	number	严重程度（1-无眼袋，2-轻微，3-严重）

名称	数据类型	说明
maskPath	String	眼袋图层图片
overallArea	object	面积
left	number	左边面积值
right	number	右边面积值
furrows	number	卧蚕（1-无卧蚕，2-有卧蚕）
wrinkles	list	细纹列表
area	number	面积值
wrinkleTypeId	number	细纹类型（1-抬头纹，2-法令纹，3-泪沟，4-笑肌断层, 5-鱼尾纹,6-眉间纹）
level	number	严重等级（1-无，2-轻度，3-中度，4-重度）
left	number	左边面积值(只有法令纹和鱼尾纹有左右面积)
right	number	右边面积值(只有法令纹和鱼尾纹有左右面积)
overallArea	object	整体面积值
left	number	左边面积值(只有法令纹和鱼尾纹有左右面积)
right	number	右边面积值(只有法令纹和鱼尾纹有左右面积)
pigmentations	list	色素斑列表
area	number	面积值
areaRatio	number	面积占比
pigmentationTypeId	number	色素斑类型（1-黑痣，2-黄褐斑，3-雀斑，4-隐藏斑）
level	number	严重等级（1-无，2-轻度，3-中度，4-重度）
facePart	number	部位（0-脸部无色素斑，1-额头，2-鼻子，3-左脸，4-右脸，5-下颌）
moisture	list	水分列表
className	number	缺水类型（1-滋润，2-敏感性缺水，3-油脂性缺水，4-老化性缺水）
level	number	严重等级（1-滋润，2-轻度缺水，3-重度缺水）
facePart	number	部位（1-额头，2-鼻子，3-左脸，4-右脸，5-下颌）

名称	数据类型	说明
sensitivity	object	敏感度
sensitivityCategory	list	敏感度类型列表
level	number	严重等级（1-无，2-轻度，3-中度, 4-重度）
facePart	number	部位（1-额头，2-鼻子，3-左脸，4-右脸，5-下颌）
sensitivityMaskPath	String	敏感图层图片
sensitivityOverall	object	敏感整脸的情况
area	number	敏感面积占比
areaRatio	number	敏感面积占比
level	number	敏感严重程度（1：无 2：轻度 3：中度 4：重度）
typeId	number	敏感类型（1-耐受，2-敏感）
skinType	number	肤质类型（1: 干性 2：中性偏干 3：中性 4：混合性偏干 5：混合性 6：混合性偏油 7：油性）
darkCircle	list	黑眼圈列表
type	number	缺水类型（1-无，2-血管型，3-色素型）
level	number	严重等级（1-无，2-轻微，3-严重）
position	number	部位（1-左眼，2-右眼）
skinAge	number	肌肤年龄
oil	list	油分列表
level	number	严重等级（1-无，2-轻微，3-严重）
facePart	number	部位（1-额头，2-鼻子，3-左脸，4-右脸，5-下颌）
fatGranule	list	脂肪粒列表
level	number	严重等级（1-无，2-轻，3-重）
number	number	脂肪粒数量
fatGranuleTypeId	number	脂肪粒类型（1-汗管瘤，2-栗丘疹）
maskPath	String	脂肪粒图层图片
wrinkleLayer	string	细纹图层图片
crowfeetMaskPath	string	鱼尾纹图层图片

名称	数据类型	说明
acneLayer	string	痘痘图层图片
pigmentationLayer	string	色斑图层图片
darkCircleMaskPath	string	黑眼圈图层图片
moistureMaskPath	string	水分图层图片
oilMaskPath	string	油分图层图片
sex	number	性别（1-男，2-女）
imageQuality	Object	图片质量
lightType	number	1-正常，2-过暗，3-曝光，4-偏光
blurType	number	0-模糊，1-正常
facePose	Object	人脸姿态估计
pitch	number	上下翻转的角度
roll	number	左右翻转的角度
yam	number	平面内旋转的角度
photographId	String	测肤记录ID
faceShelter	Object	遮挡物
exist	number	有无遮挡物（0-无遮挡物，1-有遮挡物）
types	Object	遮挡物种类
hat	number	帽子（0-没有，1-存在）
hair	number	刘海（0-没有，1-存在）
glass	number	眼镜（0-没有，1-存在）
sticker	number	鼻贴（0-没有，1-存在）
mask	number	口罩（0-没有，1-存在）
facial	number	面膜（0-没有，1-存在）
basemapPaths	Object	底图
path1	String	底图1URL,颜色为#809ED1（痘痘、色斑、黑头、脂肪粒维度可视化需要）

名称	数据类型	说明
path2	String	底图2URL,颜色为#B6D0C9（水、油、细纹维度可视化需要）
path3	String	底图3URL,颜色为#FFEECF（黑眼圈、卧蚕、眼袋、毛孔维度可视化需要）
starResult	Object	相似明星脸
name	String	明星名称
similarity	number	相似度
starPath	String	明星图片
oilOverall	Object	油分整脸的情况
area	number	油分面积
areaRatio	number	油分面积占比，ratio对应等级 1：缺油, 2：正常, 3：过多：ratio= [0.0, 2.05, 100.0]
level	number	严重等级（1：不出油 2：轻度出油 3：重度出油）
moistureOverall	Object	水分整脸的情况
area	number	缺水面积
areaRatio	number	缺水面积占比
level	number	严重等级（1：滋润 2：轻度缺水 3：重度缺水））
orgimageFaceLocation	Array	原图人脸坐标[x1,y1,x2,y2,rows,cols],包括左上角和右下角,原图大小（高X宽）

示例


```
{
    "code": 0,
    "data": {
        "furrows": 2,
        "photographId": "5f87a804bcd3340012168a4b",
        "starResult": {
            "similarity": 91,
            "name": "李溪芮",
            "starPath": "http://skintest.hetyj.com/218d5488b5102ecd7f758a
5d50b9a169.jpg"
        },
        "pore": {
            "level": 1,
```

```
            "number": 0,
            "maskPath": ""
        },
        "oilMaskPath": "",
        "oilOverall": {
            "level": 1,
            "area": 0,
            "areaRatio": 0.0
        },
        "wrinkles": [{
            "level": 1,
            "area": 0,
            "wrinkleTypeId": 6
        }, {
            "level": 1,
            "area": 0,
            "wrinkleTypeId": 1
        }, {
            "level": 1,
            "area": 0,
            "left": 0,
            "right": 0,
            "wrinkleTypeId": 2,
            "overallArea": {
                "left": 0,
                "right": 0
            }
        }, {
            "level": 1,
            "area": 0,
            "left": 0,
            "right": 0,
            "wrinkleTypeId": 5,
            "overallArea": {
                "left": 0,
                "right": 0
            }
        }, {
            "level": 2,
            "area": 2671,
            "wrinkleTypeId": 3
        }, {
            "level": 1,
            "area": 0,
            "wrinkleTypeId": 4
        }],
        "pigmentationLayer": "",
        "oil": [{
            "level": 1,
            "facePart": 2
        }, {
            "level": 1,
```

```
            "facePart": 5
        }, {
            "level": 1,
            "facePart": 1
        }, {
            "level": 1,
            "facePart": 4
        }, {
            "level": 1,
            "facePart": 3
        }],
        "skinType": 3,
        "darkCircle": [{
            "type": 2,
            "level": 2,
            "position": 1
        }, {
            "type": 2,
            "level": 2,
            "position": 2
        }],
        "eyeshape": {
            "eyelid": 1,
            "narrow": 3,
            "updown": 2
        },
        "facePose": {
            "pitch": "14.45",
            "roll": "0.08",
            "yam": "-1.59"
        },
        "faceshape": "H",
        "moistureMaskPath": "",
        "fatGranule": [{
            "fatGranuleTypeId": 2,
            "number": 0,
            "level": 1,
            "maskPath": ""
        }],
        "blackHead": {
            "level": 1,
            "number": 0,
            "maskPath": ""
        },
        "imageQuality": {
            "lightType": 1,
            "blurType": 0
        },
        "acneLayer": "",
        "eyebrow": {
            "left": 2,
            "right": 2
```

```
        },
        "darkCircleMaskPath": "http://skintest.hetyj.com/4f1cdd01320fca8d
425595bfe90e1629.png",
        "pouch": {
            "exist": 1,
            "level": 1,
            "maskPath": "",
            "overallArea": {
                "left": 0,
                "right": 0
            }
        },
        "sex": 2,
        "faceShelter": {
            "exist": 0,
            "types": {
                "hat": null,
                "hair": null,
                "glass": null,
                "sticker": null,
                "mask": null,
                "facial": null
            }
        },
        "orgimageFaceLocation": [148, 301, 1895, 2714, 4032, 2145],
        "wrinkleLayer": "http://skintest.hetyj.com/d047e9e8f84b25c45e4159
09f9db578a.png",
        "moisture": [{
            "level": 1,
            "facePart": 2,
            "className": 1
        }, {
            "level": 1,
            "facePart": 5,
            "className": 1
        }, {
            "level": 1,
            "facePart": 1,
            "className": 1
        }, {
            "level": 1,
            "facePart": 4,
            "className": 1
        }, {
            "level": 1,
            "facePart": 3,
            "className": 1
        }],
        "pigmentations": [{
            "level": 1,
            "area": 0,
            "pigmentationTypeId": 1,
```

```
            "facePart": 0,
            "areaRatio": 0.0
        }, {
            "level": 1,
            "area": 0,
            "pigmentationTypeId": 3,
            "facePart": 0,
            "areaRatio": 0.0
        }, {
            "level": 1,
            "area": 0,
            "pigmentationTypeId": 4,
            "facePart": 0,
            "areaRatio": 0.0
        }, {
            "level": 1,
            "area": 0,
            "pigmentationTypeId": 2,
            "facePart": 0,
            "areaRatio": 0.0
        }],
        "basemapPaths": {
            "path1": "",
            "path2": "http://skintest.hetyj.com/4491f27df51cac50a025f2cf0
0ddbba3.jpg",
            "path3": "http://skintest.hetyj.com/20c2ac632a629ee43384c57d1
7d430d9.jpg"
        },
        "isface": 1,
        "facecolor": "E_5",
        "crowfeetMaskPath": "",
        "sensitivity": {
            "sensitivityCategory": [{
                "level": 1,
                "facePart": 2
            }, {
                "level": 1,
                "facePart": 5
            }, {
                "level": 2,
                "facePart": 1
            }, {
                "level": 1,
                "facePart": 4
            }, {
                "level": 1,
                "facePart": 3
            }],
            "typeId": 1,
            "sensitivityOverall": {
                "level": 1,
                "area": 0,
```

```
                "areaRatio": 0.0
            },
            "sensitivityMaskPath": ""
        },
        "moistureOverall": {
            "level": 1,
            "area": 0,
            "areaRatio": 0.0
        },
        "acnes": [{
            "level": 1,
            "number": 0,
            "acneTypeId": 1,
            "facePart": 0
        }, {
            "level": 1,
            "number": 0,
            "acneTypeId": 3,
            "facePart": 0
        }, {
            "level": 1,
            "number": 0,
            "acneTypeId": 2,
            "facePart": 0
        }, {
            "level": 1,
            "number": 0,
            "acneTypeId": 6,
            "facePart": 0
        }, {
            "level": 1,
            "number": 0,
            "acneTypeId": 4,
            "facePart": 0
        }, {
            "level": 1,
            "number": 0,
            "acneTypeId": 5,
            "facePart": 0
        }],
        "skinAge": 22
    }
}
```


错误码

状态码	状态码说明	处理建议

0	请求成功	
100010100	缺少授权信息	请检查accessToken，appId，timestamp授权信息是否缺失或错误
100010101	accessToken错误或已过期	重新获取accessToken
100010103	AppId不合法	请检查是否与申请的appId一致
100010104	timestamp过期	获取最新时间戳
100010105	签名错误	请检查是否符合签名规则
100010106	请求地址错误	请检查请求地址
100010107	请求Scheme错误	请检查请求scheme是否为https
100010200	失败	未知原因，请重试
100010201	缺少参数	检查是否缺失必传参数
107001011	分析失败	图片分析失败，请重试
107001013	图片中未检测到人脸	请重新拍照
107001014	有两张或多张人脸	请重新拍照
107001032	图片太大错误	请检查图片大小是否符合要求
107001033	图片格式错误	请检查图片格式是否符合要求
107001034	图片处理超时	请重试
107001035	非法的图片路径	请重试
107001036	解析图片发生错误	请重试
107001037	图片像素未达到要求	请检查图片像素是否符合要求
107001038	维度值无效	请检查维度值是否符合文档要求
107003010	肤色检测失败	请重试
107003011	脸型检测失败	请重试
107003012	黑头检测失败	请重试
107003014	毛孔检测失败	请重试
107003016	痘痘检测失败	请重试
107003017	眼型检测失败	请重试
107003018	眉形检测失败	请重试

107003019	卧蚕或眼袋检测失败	请重试
107003020	细纹检测失败	请重试
107003021	色素斑检测失败	请重试
107003022	敏感度检测失败	请重试
107003023	油分检测失败	请重试
107003024	水分检测失败	请重试
107003025	肤质类型检测失败	请重试
107003028	脂肪粒检测失败	请重试
107003089	图片质量检测错误	请重试
107003090	人脸姿态检测错误	请重试
107003091	算法服务请求超时	请重试
107004000	授权超时	授权已超过使用期限
107005000	数据解析失败	本地解析网络数据失败