

百度人脸镜头工具使用说明

目 录

1 概述	2
2 使用前说明.....	2
2.1 百度人脸算法提供的能力	2
2.2 图像质量要求	3
2.3 图像数据回调接口.....	5
3 镜头调试工具使用操作说明.....	8
3.1 安装与操作主流程.....	8
3.2 分辨率设置	9
3.3 人脸检测设置	10
3.4 活体模式设置.....	11
3.5 活体阈值设置	12

1 概述

随着深度学习技术的发展突破，人脸识别等一系列机器视觉技术得到了长足的进步，使更多的行业痛点得以使用人工智能的方式解决。而行业的解决方案不仅涉及算法的革新，更需要新型的硬件为载体，机器视觉方向应用的落地促进了越来越多的镜头模组产品的诞生。

而百度作为一家人工智能公司，在战略上的前瞻布局使得百度机器视觉技术走到了世界的前列。为了更好得服务于社会，服务于行业，百度愿意开放自己的能力与广大镜头模组厂商组成联合生态，推进视觉算法和镜头模组硬件兼容，赋能镜头厂商具备更好的行业解决方案能力。

百度人脸镜头调试工具是一款集成了百度人脸识别算法的测试工具，旨在帮助镜头模组厂家更好地兼容和使用百度的能力，降低接入门槛。

2 使用前说明

2.1 百度人脸算法提供的能力

百度人脸算法支持基于各种2D、3D拍摄硬件的人脸检测、人脸识别和活体判断。主要采用RGB识别+3D活体判断的模式。

目前活体算法可支持RGB、IR、Depth等方式和组合，具体可包括:

组合
RGB
RGB+IR

RGB+Depth
RGB+Depth+IR
IR+Depth

2.2 图像质量要求

其中，对于三种图像要求如下：

1.RGB 图像

图像要求分辨率达到 **VGA** 及以上，图像清晰，不存在扭曲、模糊、过暗、过曝、噪点过多等情况。

最佳输入：人脸大小超过 120x120 像素，包含一定背景，人脸区域占整体图像的三分之一；

可行样例：



不可行样例：

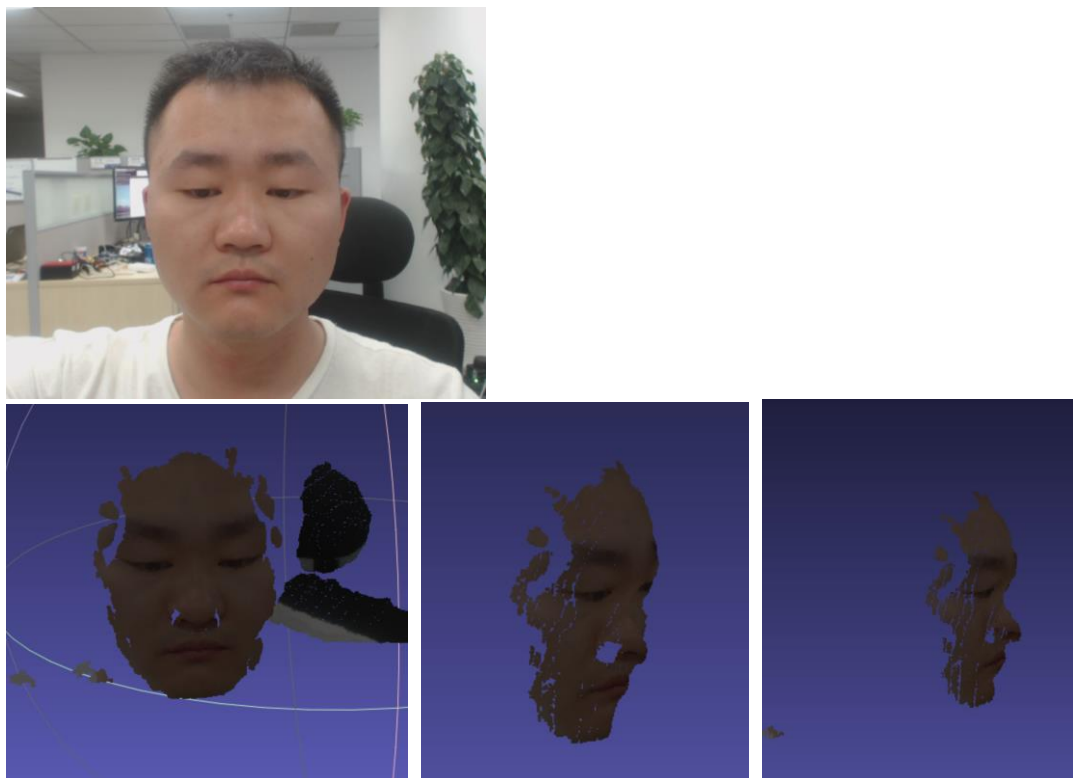


2、Depth图像

Depth图像为16位单通道图像(采集数据时可以保存为**PNG**或**TIFF**格式)，要求分辨率达到**VGA**等级及以上。其中每个像素代表对应观测点到摄像机平面的距离，单位为**mm**。其中，要求至少人脸部分有尽量连续的深度值，不存在过多的空洞(没有深度信息)。算法输入为深

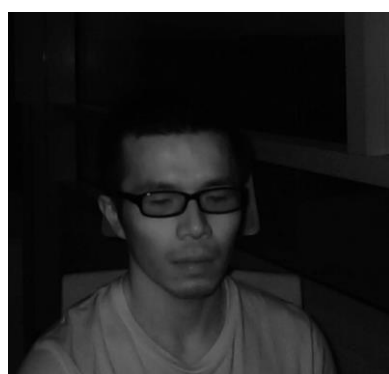
度图，以及Depth图中人脸的landmark。因此，一般要求深度图已经与RGB图像或者IR图像已对齐(或者视差很小)，此时可在RGB图像或IR图像中进行人脸检测和跟踪，从而获取Depth图中人脸的landmark信息。如果RGB图像与Depth图像没有对齐，但摄像头参数已知，也可以通过计算将RGB图中人脸的位置投影到Depth图像上。

对于模型的训练数据，测量精度最小达到距离的 0.5%，点云效果大致如下：

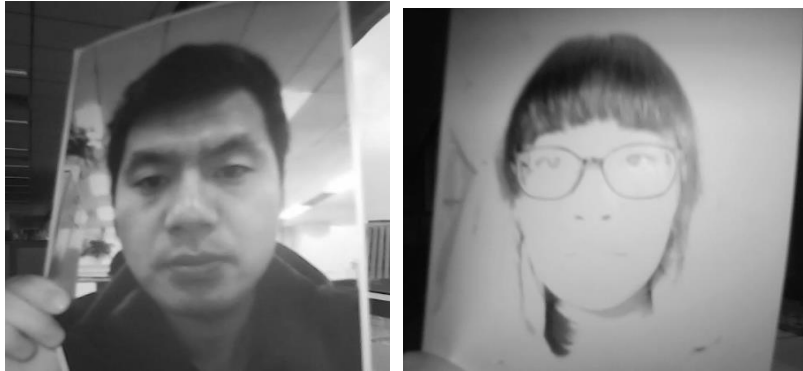


3.IR图像。要求分辨率达到 VGA 等级及以上，红外灯光谱为 850nm。算法输入为IR单通道图像(目前有些摄像头返回是转成RGB的图像,不能用于活体检测)和IR图像中landmark的位置。关于红外图有多种类型，目前inutive 模组采集的红外图片是带斑点的，这种红外图片称之为pattern，与之相对应的还有一种干净的红外图片称之为clean，目前算法检测的图片要求是clean红外图，就目前模组采集红外图片而言，Boteye采集的是clean红外；inutive采集的是pattern红外。

真人成像：



照片攻击:



屏幕攻击:



2.3 图像数据回调接口

1、RGB/IR/Depth 活体计算数据同一接口返回

活体检测逻辑要求所有组合的 (RGB/IR/Depth) 活体检测同一时刻都达到阈值才算通过 ,

所以需要RGB、IR及Depth数据在同一个回调接口返回。

2、RGB/IR/Depth活体计算数据格式

百度人脸活体算法接口说明

```
public static float run_livenessSilentPredict(int live_typeid, int[] img, int rows, int cols, int  
imgType, int[]landmark)
```

```
public static float run_livenessSilentPredict(int live_typeid, byte[] img, int rows, int cols, int  
imgType, int[]landmark)
```

参数说明 :

live_typeid 活体类型 (0 为 IR , 1 为 Depth , 2 为 RGB)

字段名称	含义	值	类型	必传
入参：				
live_typeid	静默活体图像类型	0, 红外图像 1, 深度图像 2, 可见光图像		
Img	img : 支持传入的图像数据 int[] 或 byte[]。	ARGB_8888 的像素数组	int[rows * cols]	Y
		IR 为单通道 8bit Depth 为 unsigned short 16bit	byte[]	
rows	图像的高	height	int	Y
cols	图像的宽	width	int	Y
imgtype	图像类型(标记位数)	0, 红外图像 2, 深度图像 24, 可见光图像	int	Y
landmark	人脸特征点坐标数组	faceInfo.landmarks	int[]	Y
出参：				
参数数据类型 float , 表示活体检测分数, 值范围[0-1],异常返回-1				

以上上人脸算法的原子接口。用于封装多种活体的组合。

3、百度人脸调试工具示例代码中的封装接口说明与使用：

A. 代码中对人脸检测进行了封装FaceDetectManager类中,增加了回调机制。使用方法：

```
faceDetectManager.setOnFaceDetectListener(new
FaceDetectManager.OnFaceDetectListener({
    @Override
    public void onDetectFace(int retCode, FaceInfo[] infos, ImageFrame frame) {
        // 显示检测的图片。用于调试, 如果人脸sdk检测的人脸需要朝上, 可以通过该图片判断
    }
});
```

B. 代码中对活体检测进行了封装FaceLiveness类中, 增加了回调机制。使用方法：

```
FaceLiveness.getInstance().setRGBInt(RGBData); //设置RGB数据
FaceLiveness.getInstance().setIrData(irData); //设置近红外数据

FaceLiveness.getInstance().setDepthData(DepthByte); //设置深度数据

//活体检测封装的方法
FaceLiveness.getInstance().setLivenessCallBack(new ILivenessCallBack() {
    public void onCallback(LivenessModel livenessModel) {
```

```

        livenessModel.getRgbDetectDuration() //检测耗时
        livenessModel.getRgbLivenessDuration() //RGB活体耗时
        livenessModel.getRgbLivenessScore() //rgb活体得分
        livenessModel.getIrLivenessDuration() //Nir活体耗时
        livenessModel.getIrLivenessScore()//Nir活体得分
        livenessModel.getDepthLivenessScore() //depth活体得分
        livenessModel.getDepthLivenessDuration() //depth活体耗时
    }
    public void onTip(int code, String msg) {

    }
    public void onCanvasRectCallback(LivenessModel livenessModel) {

    }
});

```

C.硬件厂商数据回调接口要求

- 1) RGB数据类型 : int[]或byte[] c4u8
- 2) IR数据类型 : byte[] c1u8
- 3) Depth数据类型 : byte[] unsigned short[转换的byte[]

活体计算数据回调接口类似onDataCall , RGB必选 , IR、 Depth根据摄像头模组能力可选

```

public void onDataCall(int[]RGBImageBytes, intRGBWidth, intRGBHeight,
                    byte[]DepthDataBytes, intDepthWidth, intDepthHeight,
                    byte[] irImageBytes, int irWidth, int irHeight)
public void onDataCall(byte[]RGBImageBytes, intRGBWidth, intRGBHeight,
                    byte[]DepthDataBytes, intDepthWidth, intDepthHeight,
                    byte[] irImageBytes, int irWidth, int irHeight)

```

显示回调接口 , 如果计算接口返回帧率达不到15+ , 需要提供显示帧率15+的接口 (RGB是

用于用户交互 , IR和Depth用调试 , 帧率要求不高)

```

public void onRGBDisplayCall(byte[]RGB, int width, int height)
public void onNIRDisplayCall(byte[] ir, int width, int height)
public void onDepthDisplayCall(byte[]Depth, int width, int height)

```

3 镜头调试工具使用操作说明

3.1 安装与操作主流程

- a.安装环境推荐使用：RK3399 主板+Android5.0 以上系统+AndroidStudio 开发工具
- b.设置好硬件环境，连接上镜头，在 AndroidStudio 安装百度提供的调试工具，并运行
- c.工具主界面如下，先在四个设置按钮里设置好相应参数，点击“开始测试”，若接入没有问题，即可看到输出结果

主界面：



输出结果 (例):



3.2 分辨率设置

分辨率默认为 1080p、720p、480p 三种，每种图片类型，只能选择一个分辨率。



3.3 人脸检测设置

人脸检测设置：进入主页面，选择人脸检测设置进入，可以进行设置最小检测人脸的大小

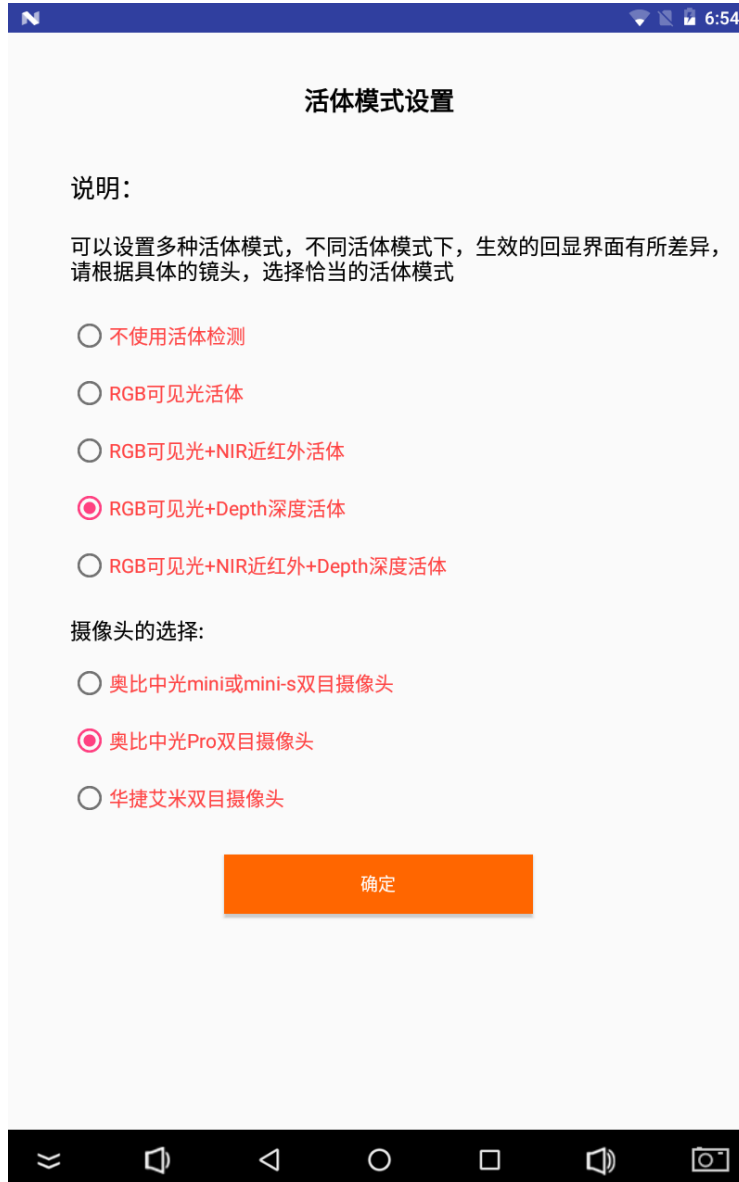
最小检测人脸默认 60，每次增减 20，取值范围：60、80、100、120、140、160。

关键点点击「开启」，则在 RGB，及 NIR 或者 Depth 回显上，打印出来关键点，以 RGB 中的关键点为基准。



3.4 活体模式设置

活体模式设置：进入主页面，选择活体模式设置进入，可以进行设置需要测试的活体类型和提供的一些现有的镜头。包括无活体，RGB 可见光，RGB+NIR，RGB+Depth。现有的深度的包括奥 mini/Pro 镜头，华捷艾米镜头。



3.5 活体阈值设置

活体阈值设置：进入主页面，选择活体阈值设置进入，可以进行设置 RGB 活体阈值，NIR 活体阈值，Depth 活体阈值。默认值是 0.90。测试的时候活体返回大于等于所设置阈值才算通过。

