## 涉及第三方库：

1. Opencv 4.4
2. Onnxruntime 1.13.1

## 交付结构：

1. 算法项目

.h + dll （在.h中进行传参）

.h中传入图片所在文件夹路径，这个文件夹中有三张图片是我们需要的。传入图片名称在.h文件中写死。涉及模型名称在.h文件中写死。后续正式传参只传文件夹路径。

1. 检测项目

.h + dll （在.h中进行传参）

.h中传入图片的绝对路径，一共就两个函数，一个inface 一个 faceslice。

2.1 Inface传入图片，返回两个值1和0； 1标识图片合规，0标识不合规。

2.2 faceslice传入图片和保存图片的路径，返回一个图片写到指定路径，返回的就是一个边缘检测的无底色图。

## 设计接口：

传入一个路径（这个路径中只包含用到的三张图片路径），在.h文件中，预先设定好模型名称以及需要用到的图片名称。

## 返回结果：

将返回结果存到传入的内个路径文件夹中(同时需要把每个图像缩略两份放到两个文件夹中（mid文件夹：280\*420），（samll文件夹：120\*180）)。

输出结果图片：

1. Red.jpg （红图）
2. Brown.jpg （棕图）
3. Reli.jpg （热力图）
4. Qushi.jpg （趋势图）
5. You.jpg （紫图）
6. BW.jpg （黑白图）
7. LV.jpg （绿图）
8. OLD.jpg （老化图）
9. Red\_Crop.jpg （红图色素分割）
10. Brown\_Crop.jpg （棕图色素分割）
11. Reli\_Crop.jpg （热力色素分割）
12. You\_point.jpg （紫图点识别）
13. BW\_Crop.jpg （黑白图分割）
14. LV\_Crop.jpg （绿图分割）
15. Face\_ZW.jpg （原图皱纹检测）
16. Face\_MK.jpg （原图毛孔检测）

## 模型、算法与图片对应：

1. img\_cross.jpg ：Red.onnx、Brown.onnx、Reli.onnx、Qushi.onnx、FaceCrop.onnx（分割人脸区域）
2. Img\_uv ：You.onnx、BW.onnx、LV.onnx
3. img\_hor.jpg ：OLD.onnx、ZW、MK

## 后处理：

全部色图（紫图除外）色素分割

紫玉识别关键点

原图检测皱纹和毛孔黑头 然后组合输出

在计算每个色素分割在facecrop的占比

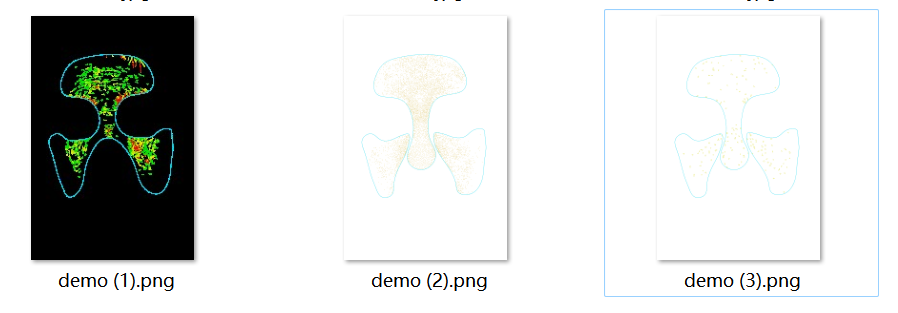
## 模型组调用流程：

1. 传入一个文件夹路径，这个文件夹中有咱们需要的三张图片img\_cross.jpg, img\_uv.jpg, img\_hor.jpg。咱们算法模型所需要的全部图片就是这三张。
2. 让img\_cross.jpg分别传入Brown.onnx、Red.onnx、Reli.onnx、Qushi.onnx、FaceCrop.onnx这5个模型，输出四张图片和1个实例分割mask的结果。

大概类似这种：



有了这个mask部分，咱们就可以基于其他算法去获取最终图。在他基础上应用算法后效果类似这种（注意这个图咱们是要mask部分的边界。方框忽略掉）：



1. Img\_uv.jpg传入You.onnx、BW.onnx、LV.onnx这三个模型。然后模型输出You.jpg、BW.jpg、LV.jpg这三个图
2. Img\_hor.jpg传入OLD.onnx模型中，输出OLD.jpg图片。
3. Img\_hor.jpg调用ZW（皱纹描画）、MK（毛孔检测【这个我还没做出来】）这两个算法接口，然后再配合FaceCrop分割出的区域输出Face\_ZW.jpg图片，效果类似这样：



1. 然后基于2.中输出的

|  |
| --- |
| Brown.jpg |
| Red.jpg |
| Reli.jpg |
| You.jpg |
| BW.jpg |
| LV.jpg |

这几张图片，分别调用Crop、Crop、Crop、Point、Crop这五个算法（依然要结合FaceCrop限定的区域）输出图片：

|  |
| --- |
| Brown\_Crop.jpg |
| Red\_Crop.jpg |
| Reli\_Crop.jpg |
| You\_Point.jpg |
| BW\_Crop.jpg |
| LV\_Crop.jpg |

# 注意：

# 所有输出的算法图用png格式。我刚刚写错了。FaceCrop这个可以不单独输出图片。

# 所有FaceCrop参与的图片都要无底色的透明图。并且所有输出的图片，都写调用接口时候传入的文件夹路径中。

## 两个简易算法的接口模型：

.h文件中定义两个接口函数。一个inface 一个 faceslice。这两个接口是没有关联的。

Inface是传入一张图片，函数最终返回1或者0。

Faceslice也是传入一张图片和一个文件夹地址。函数返回一张图片，写到传入的文件夹地址中。