

这是一个通用的机器学习算法结果展示程序，通过 socket 与 机器学习程序通信，让研究人员无需编写任何代码，就能方便的演示 算法效果。

优点：具有点击图片放大，自适应图片大小，能够演示机器学习程序即时标记视频。全平台，支持 Ubuntu Windows7 以上。

Windows 双击 exe 运行，linux 直接把 run 文件拖入终端运行(如果运行失败请 chmod +x 赋予执行权限)

先运行本程序，当用户点击按钮，选择文件之后，程序会将文件路径传给你所编写的机器学习程序，机器学习程序预测之后将标记的图片和预测结果传给本程序，本程序会进行显示。

整个过程，机器学习研究员只需写三行代码。研究员无需关心任何界面细节。

1.将 UniversalGUI.py 放在你工作目录

2.在你的程序里 **import \* from UniversalGUI**

3.运行展示程序

(img 为 opencv 格式 BGR 通道)

**sendOriginImg(img)** #这行代码将会把图片显示在展示程序原始图片区域

**sendSignedImg(img)** #这行程序将会把图片显示在标记图片区域

**sendResult("识别结果：\n 当前液面 10%")**

# 这行代码将会把当前帧预测的结果发送给演示程序，演示程序会立即显示

你需要编写两个函数

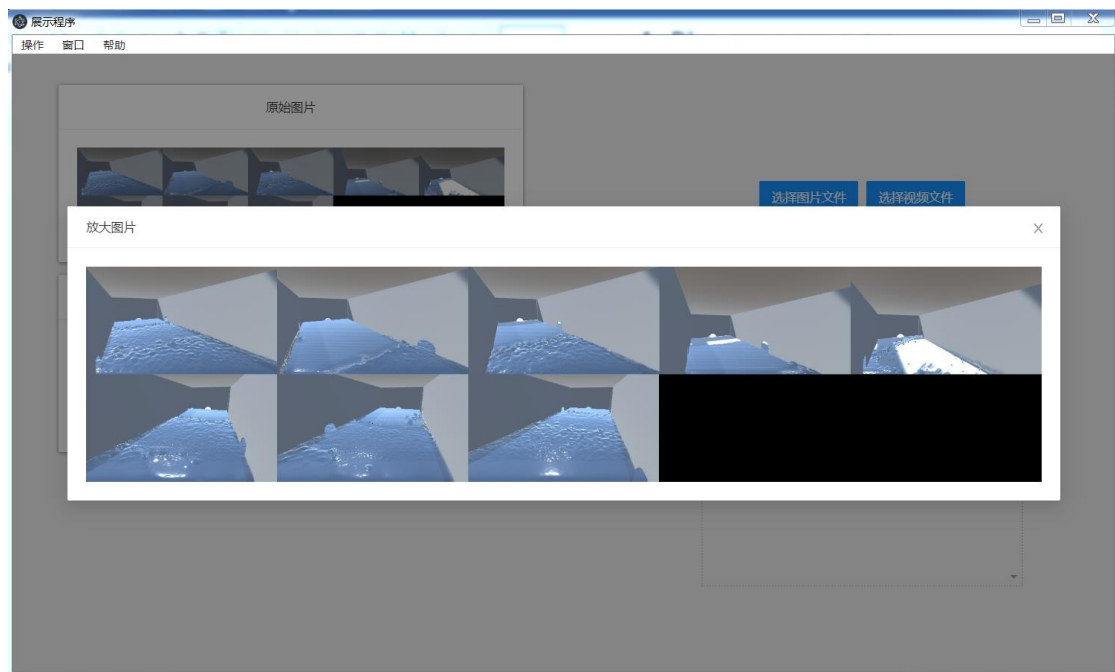
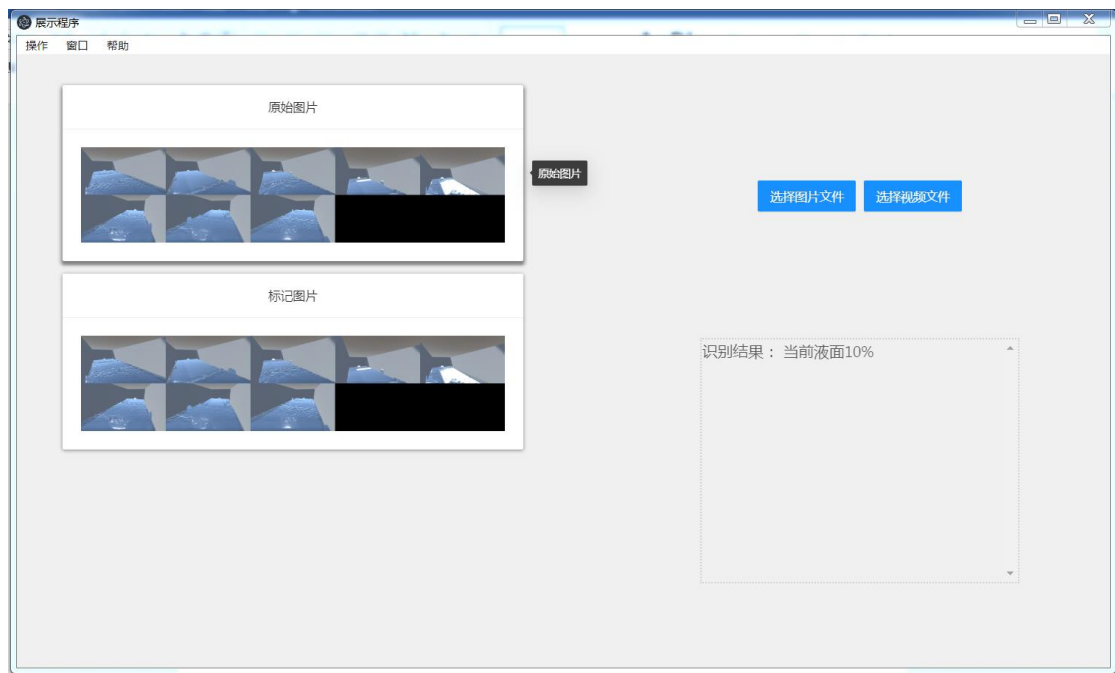
**whenUserChooseImage(imagePath)** #当用户点击选择图片的时候这个函数会被自动执行，imagePath 即 是用户选择的图片路径，你在这里调用你的机器学习程序预测图片，然后用前文的 send 函数来显示预测结果。

**whenUserChooseVideo(VideoPath)** #当用户点击选择视频的时候这个函数会被自动执行，VideoPath 即 是用户选择的视频路径，你在这里调用你的机器学习程序循环预测视频每一帧，然后用前文的 send 函数来显示预测结果和当前标记的帧。

**waitForUserClickChooseFils(whenUserChooseImage,whenUserChooseVideo)**

将你编写得的两个函数填入 waitForUserClickChooseFils 中，（不要加括号），waitForUserClickChooseFils 会阻塞程序，等待用户的操作来自动调用机器学习程序，预测结果返回给展示程序。

这个帮助文件也可以在展示程序的帮助菜单中打开



示例程序:

```

if __name__ == '__main__':
    def whenUserChooseImage(ImagePath): # 此函数将会在用户选择完图片之后被自动运行
        # ImagePath 为用户选择的图片路径
        img = cv2.imread(ImagePath)
        sendOriginImg(img) # 这行代码将会把图片显示在展示程序原始图片区域
        #
        # 你在这里调用机器学习程序预测结果并且标记好图片
        #
        sendSignedImg(img) # 这行程序将会把图片显示在标记图片区域

    def whenUserChooseVideo(VideoPath): # 此函数将会在用户选择完视频之后被自动运行
        # VideoPath为用户选择的文件路径
        img = cv2.imread(VideoPath)
        #
        # 机器学习程序在这里循环读取视频每一帧
        # 机器学习程序 预测这个视频的每一帧，并且标记好图片
        #
        sendSignedImg(img) # 这行代码将会把标记的帧发送给展示程序，展示程序会立即显示
        sendResult("识别结果：\n 当前液面10%") # 这行代码将会把当前帧预测的结果发送给演示程序，演示程序会立即显示
        #
        # 视频预测完毕循环结束
        #

    waitForUserClickChooseFiles(whenUserChooseImage, whenUserChooseVideo)
    # 将上方编写的两个函数填入这个函数的参数内，
    # 只填入函数名称不要加括号（因为在这里这两个函数本质上是回调函数）
    # 这个函数会立即阻塞程序进程，等待用户的操作来自动调用机器学习程序，预测结果返回给展示程序

```