基于智慧教室的考场异常行为检测系统

使用手册

**目录**

[一、系统修订记录 2](#_Toc160463535)

[二、系统简介 2](#_Toc160463536)

[三、系统环境配置 2](#_Toc160463537)

[3.1系统开发环境 2](#_Toc160463538)

[3.2系统运行环境 3](#_Toc160463539)

[四、系统功能和使用 4](#_Toc160463540)

[4.1系统运行和启动 4](#_Toc160463541)

[4.2功能使用介绍 7](#_Toc160463542)

[五、系统设计逻辑 18](#_Toc160463543)

[六、合作与开源 20](#_Toc160463544)

# 系统修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 生成日期 | 作者 | 修订内容 |
| V1.0 | 2024-3-3 | 谢骏鑫 陈岩 陈秋宇 汤嘉彧 | 发布初始版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 系统简介

在当今的教育过程中，考试作为选拔人才的最重要且权威的方式，其公平性显得更为突出，而为了充分保证考试的权威性和公平性，杜绝考试过程中的作弊现象就尤为关键了。而在当今科学技术飞速发展的支持下，智慧教室愈来愈出现在各个学校中，基于智慧教室提供的高清监控视频，我们可以通过人工智能和视频分析的前沿计算机技术，对于考试过程中的一些动作幅度较大的典型异常行为进行自动识别和抓捕，从而尽可能地省去人力资源在考后一分一秒审查考试监控视频上的消耗。

本系统正是基于这样的场景，从这个目的出发，结合人工智能中骨骼点识别、姿态估计等新兴技术，开发出一个可以检测出考试过程视频中的异常行为的发生时刻的系统应用，从而减少传统人工监考所需要耗费的人力和物力资源。

# 三、系统环境配置

## 3.1系统开发环境

|  |  |
| --- | --- |
| 操作系统 | Windows10/Windows11 |
| CPU | AMD Ryzen 9 7940H |
| 显卡 | NVIDIA GeForce RTX 4060 Laptop GPU |
| 开发语言 | Python（3.8） |
| 运行内存 | 16G |

## 3.2系统运行环境

为正常运行本系统项目，请用户确保自己的计算机配置和上述系统开发环境相差不至过大，并保证本地计算机具有NVDIA GPU。以下是为正常运行本系统所需要的一些关键包，运行前请保证下述三方包均已成功安装到本地Python运行环境中。

|  |  |  |
| --- | --- | --- |
| 安装包 | 版本 | 安装方式 |
| python | 3.8.18 | 前往python官网<https://www.python.org/downloads/>，自行安装到本地 |
| torch | 2.1.1 | 前往pytorch官网[PyTorch](https://pytorch.org/)，根据操作系统环境，安装制定cuda版本下的pytorch。  以windows11操作系统，cuda版本11.8为例，在本地终端命令行窗口中输入如下指令：pip install torch==2.1.1 torchvision==0.16.1 torchaudio==2.1.1 --index-url https://download.pytorch.org/whl/cu118 |
| torchvision | 0.16.1 |
| cuda | 11.8 | 前往<https://developer.nvidia.com/cuda-toolkit-archive>，根据所安装的pytorch版本安装匹配版本的cuda。 |
| cudNN | 8.9.7 | 前往<https://developer.nvidia.com/cudnn-downloads>，根据cuda版本号安装匹配版本的cudNN。 |
| tensorflow | 2.13.0 | 以windows11为例，在终端命令行窗口中输入：pip install tensorflow-gpu==2.13.0 |
| tensorflow-hub | 2.13.0 | 以windows11为例，在终端命令行窗口中输入：pip install tensorflow-hub==2.13.0 |
| numpy | 1.24.3 | 以windows11为例，在终端命令行窗口中输入：pip install numpy |
| opencv-python | 4.9.0.80 | 以windows11为例，在终端命令行窗口中输入：pip install opencv-python |
| pandas | 2.0.3 | 以windows11为例，在终端命令行窗口中输入：pip install pandas |
| Pyqt5 | 5.15.10 | 以windows11为例，在终端命令行窗口中输入：pip install PyQt5 |
| requests | 2.31.0 | 以windows11为例，在终端命令行窗口中输入：pip install requests |
| pillow | 10.0.1 | 以windows11为例，在终端命令行窗口中输入： pip install pillow |
| matplotlib | 3.7.4 | 以windows11为例，在终端命令行窗口中输入：pip install matplotlib |
| scipy | 1.10.1 | 以windows11为例，在终端命令行窗口中输入：pip install scipy |
| pickle |  | 以windows11为例，在终端命令行窗口中输入：pip install pickle |
| collection |  | 以windows11为例，在终端命令行窗口中输入：pip install collection |
| scikit-learn | 1.3.2 | 以windows11为例，在终端命令行窗口中输入：pip install scikit-learn |

需要说明的是，上述Python三方包的版本和安装示例均为参考开发过程的环境进行指导，并适当省略了部分所需的安装教程，若在安装过程中出现报错，用户可首先检查本地计算机配置是否达到需求，其次推荐在互联网搜索报错信息寻找对应的解决方案。

此外，为了保证用户上传的视频可以在系统应用界面中正常播放，还需要再本地系统上安装一个解码器，推荐安装K-Lite Codec Pack Basic解码器，以安装该解码器为例，首先进入解码器官网：<https://www.codecguide.com/download_k-lite_codec_pack_basic.htm>，选择“Download”下的“Server 1”点击下载即可，考虑到可能会出现部分浏览器对于陌生程序的下载拦截，下载完成后在浏览器的下载器中对此程序选择信任并保留，然后执行安装过程即可，安装流程一直选择“next”直至安装成功。

# 四、系统功能和使用

## 4.1系统运行和启动

在运行和启动本系统应用之前，确保您的设备所处的运行环境已经达到上述第三点所描述的要求，包括但不限于计算机设备的CPU级别、显卡硬件级别、内存级别、运行所处Python语言的环境要求等，若无法顺利运行系统项目，请务必检查第三点所要求的运行环境配置是否达到、是否正确配置。

如下图4-1所示，这是本系统项目使用PyCharm集成编辑器打开本系统项目后的界面分布情况，项目各文件的组织情况如图左边项目组织视图所示，文件组织中找到“qt\_index.py”的文件，这是本系统项目启动主页面的主文件，在PyCharm中鼠标右键此文件选择运行，若上述环境要求均已达到并且配置正确，稍后片刻后将弹出使用本系统应用的提示弹窗，如下图4-2所示，点击“OK”后将进入本系统项目的主页面，主页面如图4-3所示。提示弹窗建议用户在使用前阅读菜单栏中的“用户使用手册”，其中包含了使用本系统软件的注意事项。

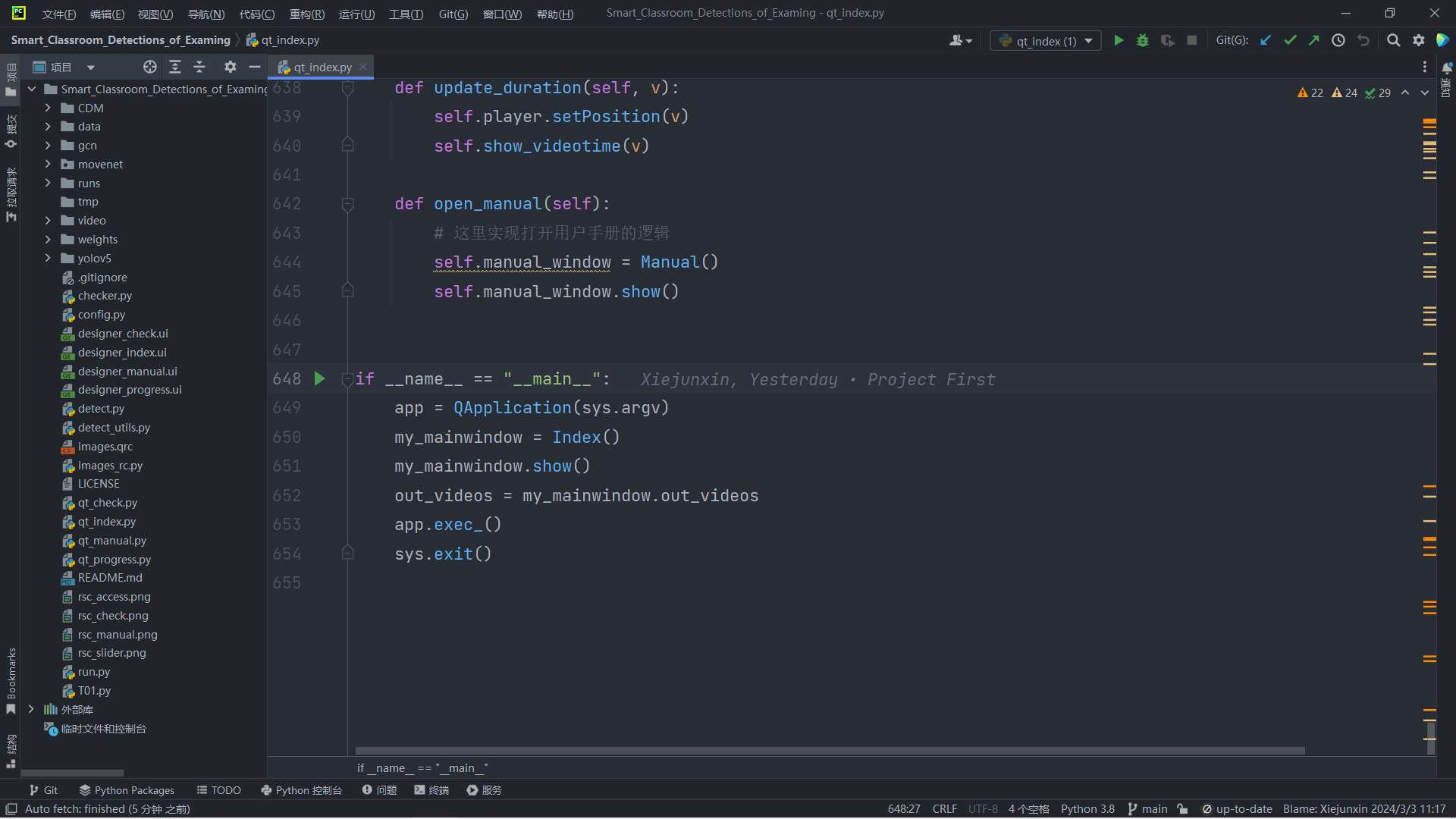


图4-1 PyCharm打开本项目的分布

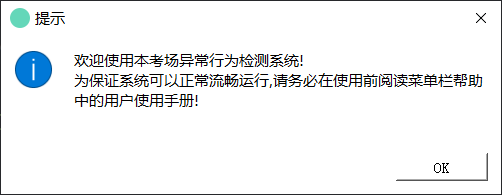


图4-2项目顺利启动前的提示弹窗



图4-3基于智慧教室的考场异常行为检测系统的主程序页面

需要注意的是，使用手册这里仅仅是使用PyCharm这一集成开发编辑器作为示例，事实上，任一Python语言的集成开发环境均可以启动本系统项目，当您的计算机设备中正确配置了上述所需的环境，您甚至可以在本地的命令行程序启动本项目，这里以Windows10操作系统为例进行演示。

首先，本系统项目的根目录防止在本机系统的D盘上，文件路径为“D:\ Smart\_Classroom\_Detections\_of\_Examing”，打开Windows系统的命令提示符程序cmd，可以从系统程序找到cmd并打开，也可以使用快捷键Windows徽标键win+R打开“运行”输入cmd后打开，打开后的cmd界面如下图4-4所示。

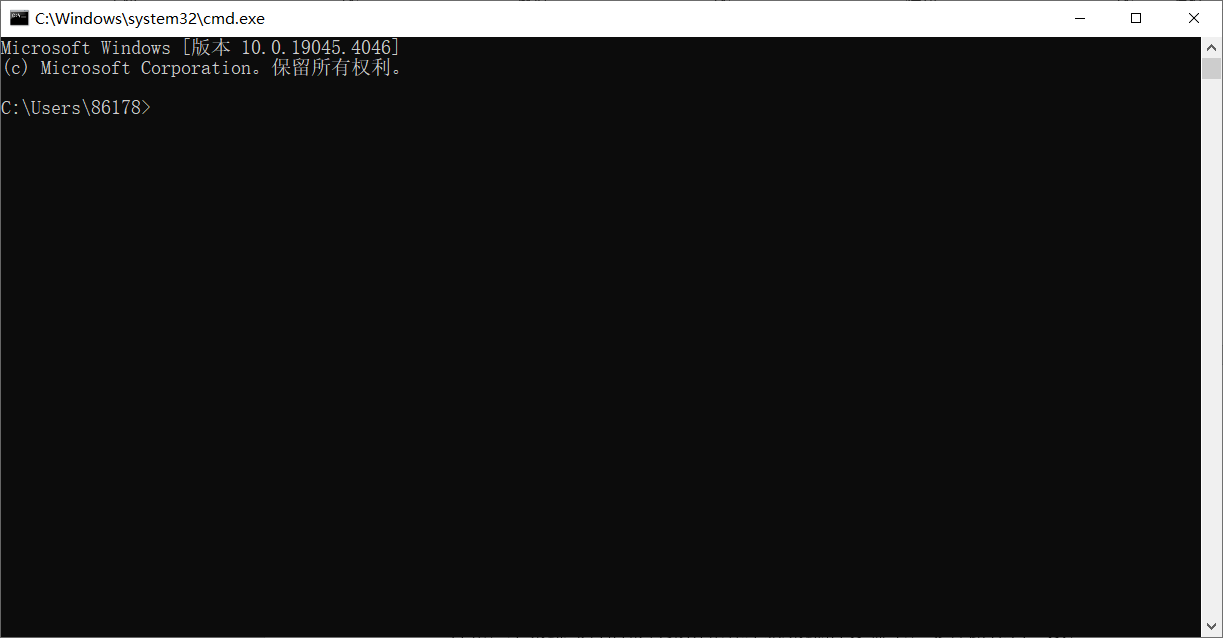


图4-4 Windows系统的cmd程序

然后将程序运行的路径更换为本系统项目所处的目录即可，以演示的路径为例，键盘键入“D:”后进入系统D盘目录下，然后输入“cd Smart\_Classroom\_Detections\_of\_Examing”进入本系统项目所处的目录下，最后执行Python运行程序的命令，输入“python qt\_index.py”后键入回车即可启动本项目进入如图4-3所示的界面，启动输入的流程如下图4-5所示。需要注意的是，使用系统的命令提示符程序启动项目，需要确保对应Python语言的环境已经被配置到系统的环境变量中，如何配置Python语言的环境变量互联网上均有大量适用的教程，根据自己所处的操作系统环境进行配置即可，这里不再赘述。

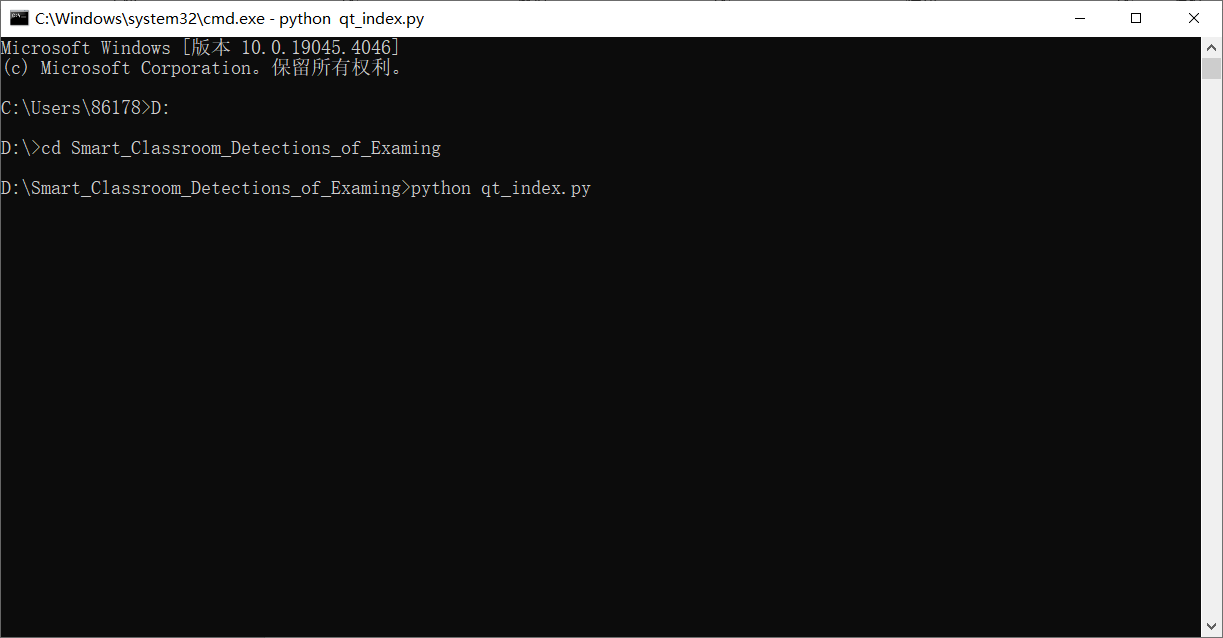


图4-5 Windows系统cmd程序启动本项目

## 4.2功能使用介绍

顺利进入如上图4-3的系统程序主界面之后，就可以正常使用本系统功能了。如上图4-3所示，主页面的布局一目了然，左上角的名称“考场异常行为检测系统”点明了本系统的名称和主要功能，下一行菜单栏中有一个“帮助”栏，此栏点击后可以查看开发组为保证用户顺利使用本系统而撰写的用户使用手册，如下图4-6、4-7所示。



图4-6主页面点击菜单栏的“帮助”按钮展开“用户使用手册”



图4-7点击“用户使用手册”后呈现的界面

在主页中心布局的左上一大部分空白安放的是界面的播放器，用户选择视频上传后选中的视频可以展示在这里进行播放。在播放器的下面有一个进度条和播放时间显示框，进度条用来展示视频播放的进度，并且用户可以通过拖拽进度条来任意定位视频播放的进度，而一旁的播放时间显示框将会在视频播放时显示当前视频正在播放的时刻。而其下方的两个黄色按钮“播放”和“暂停”的功能与其名字相一致，点击“播放”将会使暂停的视频继续开始播放，而点击“暂停”按钮将会使正在播放的视频暂停播放。视频播放功能的使用如下图4-8所示。



图4-8系统视频播放功能的使用

最下方的“选择视频源”按钮将会弹出一个文件对话框以让用户选择要进行处理和检测的视频，处理视频支持一般的视频格式，包括但不限于“.mp4”、“.avi”等格式。用户在使用本系统的核心功能“检测异常行为”前必须通过“选择视频源”上传一个正确的视频文件格式，若没有上传视频或者上传了错误的文件格式，系统均会弹出提示并请求用户重新进行操作。“选择视频源”的相关操作和提示如下图4-9、4-10、4-11所示。

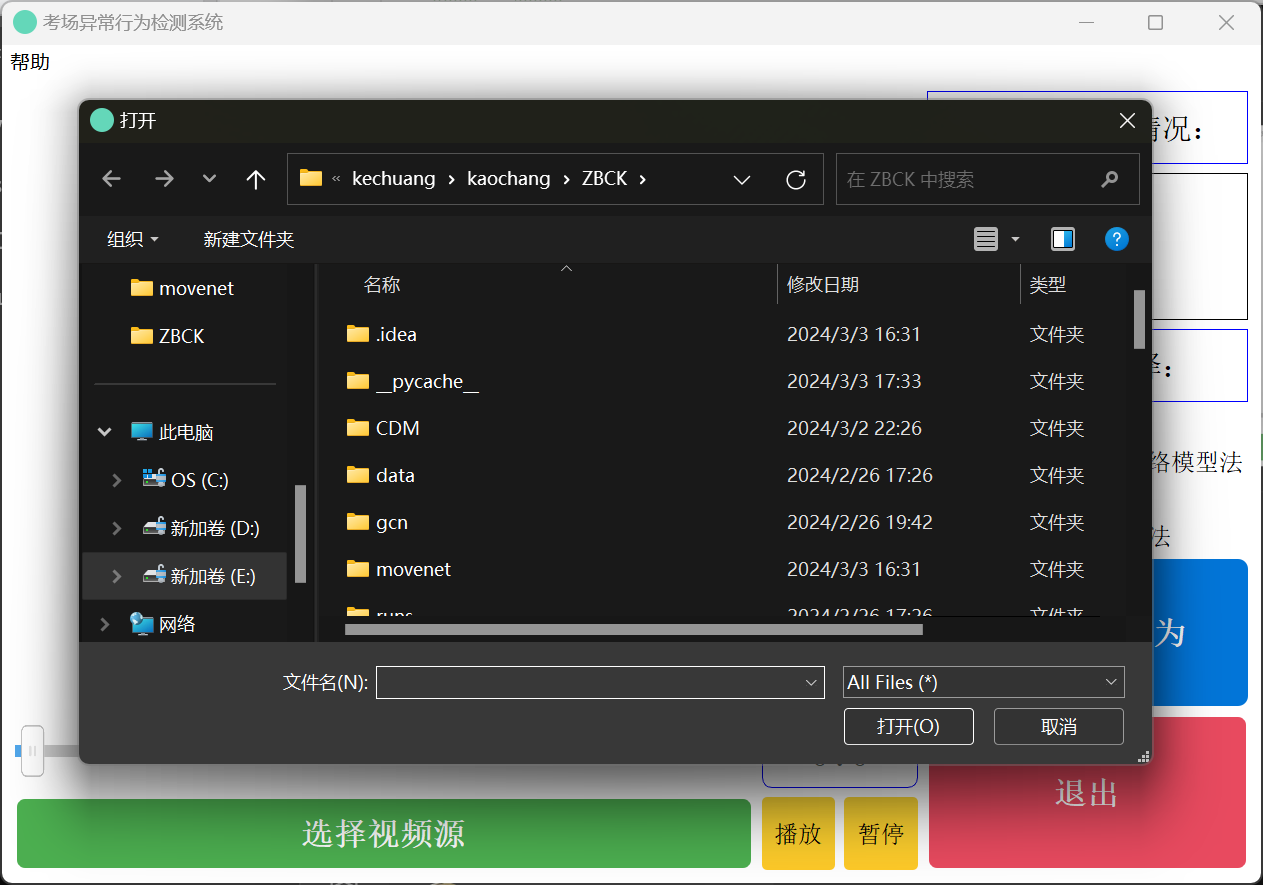


图4-9点击“选择视频源”后从本地选择一个合适的视频文件



图4-10上传错误的视频文件后的提示



图4-11没有成功上传合适的视频文件仍然选择处理后的提示

在布局的右侧部分，最上面有一个标签提示“异常行为检测情况”，并且在它的下方有一个文本提示框，这一部分将会在系统处理完成上传的视频之后显示系统对所处理的视频所检测到的异常行为出现的时刻点，以方便用户进行快速定位和查证。紧接着在它的下方显示有“处理方法选择”的信息标签，然后其下方有四个复选框以供用户选择，本系统在设计时充分为用户保留了方法使用的自由度，系统设计上采用了集成学习的方式来进行异常行为的检测，并且一个设计了四种方案，按照图中复选框的顺序依旧是：通过视频中人物的骨骼点坐标进行判断是否是异常行为、通过对骨骼点坐标进行神经网络模型训练来预测判断是否是异常行为、通过图片识别分类的方法判断当前的动作图片是否是异常行为、通过时序网络模型检测视频过程中是否产生了异常行为，用户在每次处理的过程中可以任选四种方法中的几种方法进行组合来比对各种方案的处理效果，但请务必保证至少选中一个方法，否则将会弹出提示以让用户重新操作，如下图4-12所示。

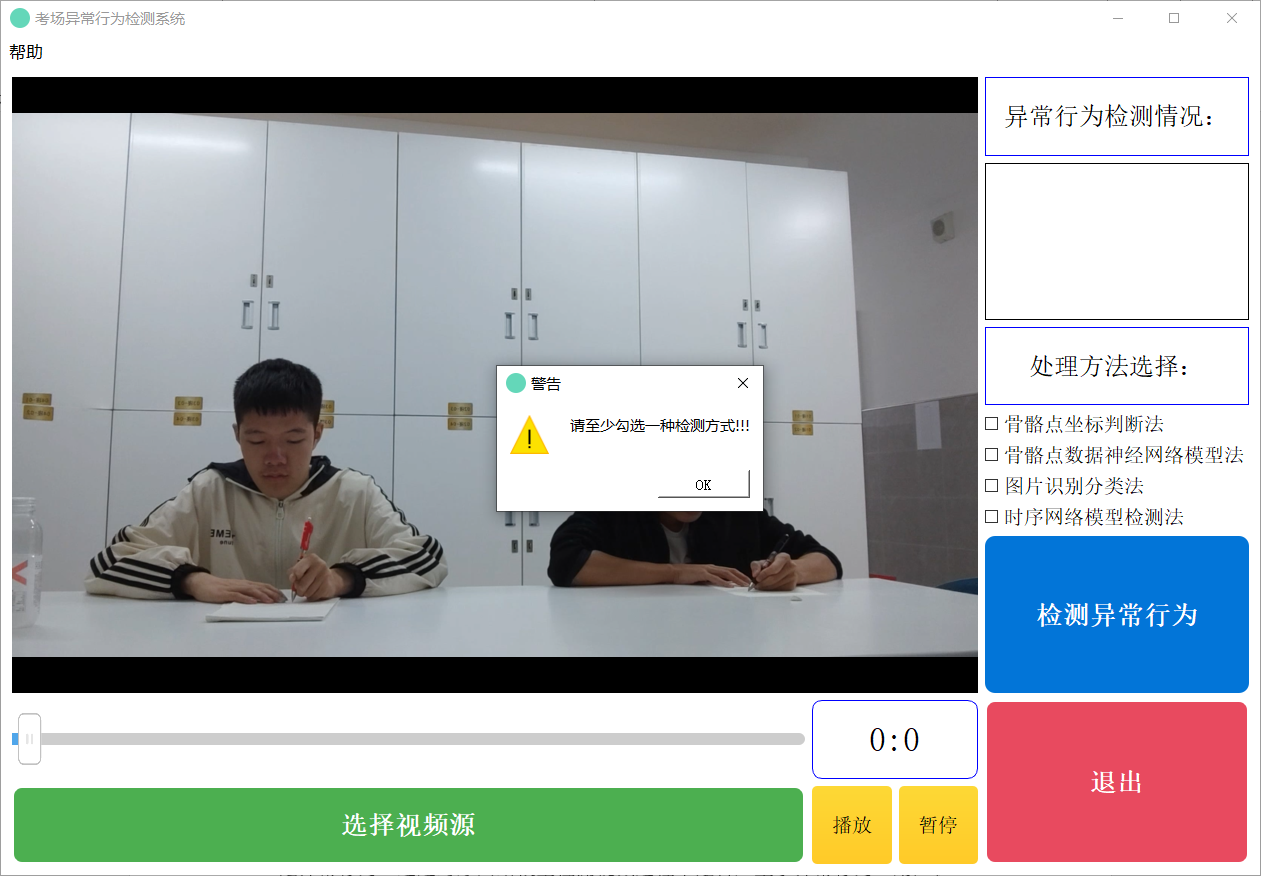


图4-12未勾选检测方式尝试检测后的提示

在正确上传视频和勾选了一个组合方案之后，点击蓝色的按钮“检测异常行为”便可以使用本系统最核心的功能，点击该按钮之后系统后台将会使用用户所勾选的方法对上传的视频进行处理，去检测视频中可能存在的异常行为，系统在此刻将会弹出一个进度条弹窗，时刻显示现在系统程序处理视频的进度，当进度条达到100%时说明视频处理完成，将进入第二个窗口页面，用户将在这个界面查看视频处理之后的效果。同时系统应用保留了在处理视频过程中每帧的处理效果，在进度条加载的过程中将会以另一个弹窗的方式显示，方便用户在等待的过程中也可以实时看到程序处理的效果，避免了干燥的等待。另外，考虑到用户可能上传了错误的视频或者勾选的检测方案并非自己想要的，在载入进度条的过程中可以点击进度条窗口中的“取消”按钮中断处理过程重新进行操作。由于每一次系统程序处理视频所需要占用大量的计算资源，因此系统增加了对多线程同时处理多个视频的限制，当用户点击“检测异常行为”之后若未取消处理过程或者关闭第二个窗口界面，主窗口的此按钮将会取消交互性，用户无法再次进行点击。相关效果图如图4-13、4-14、4-15、4-16所示。

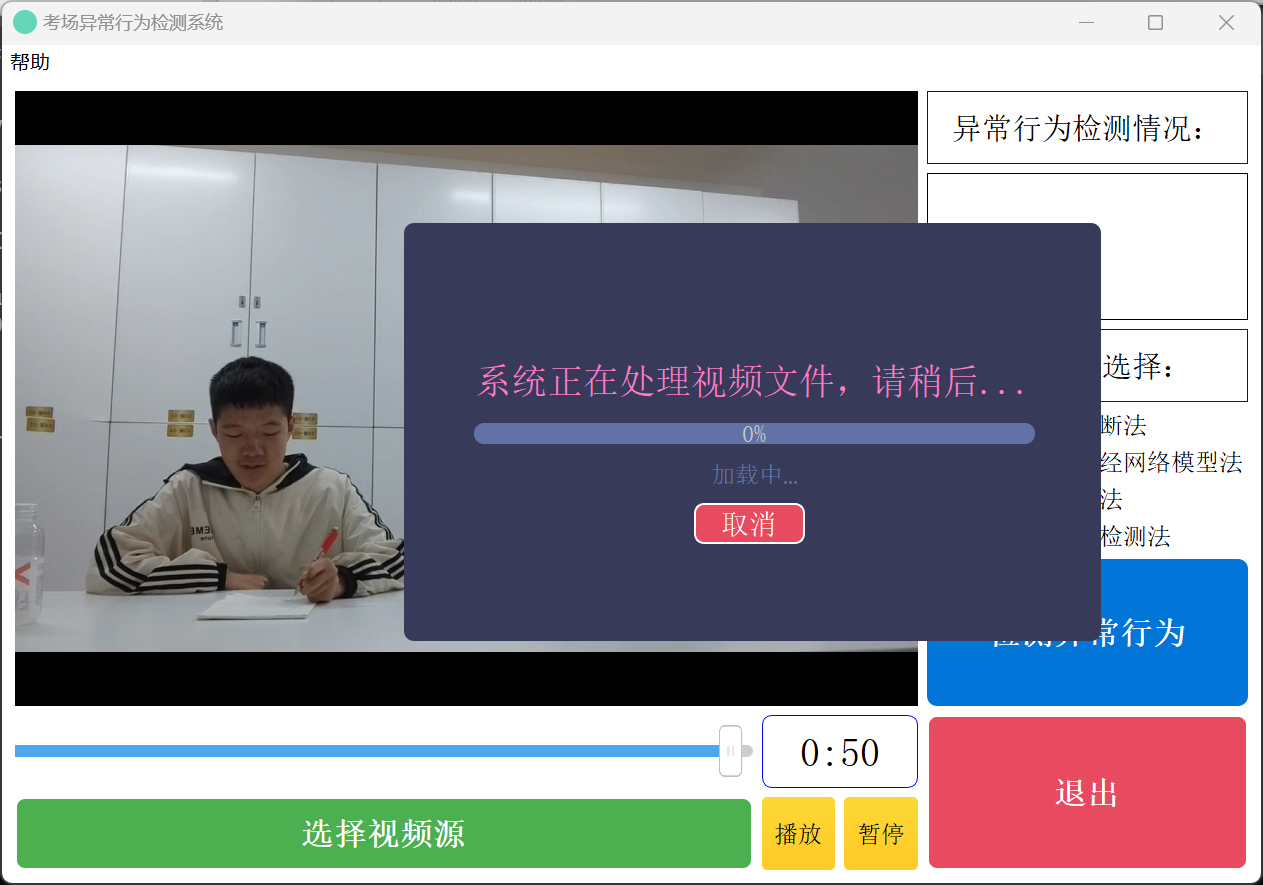


图4-13点击“检测异常行为”按钮后系统开始处理视频并弹出进度条提示



图4-14系统正在处理视频过程的进度条正确提示

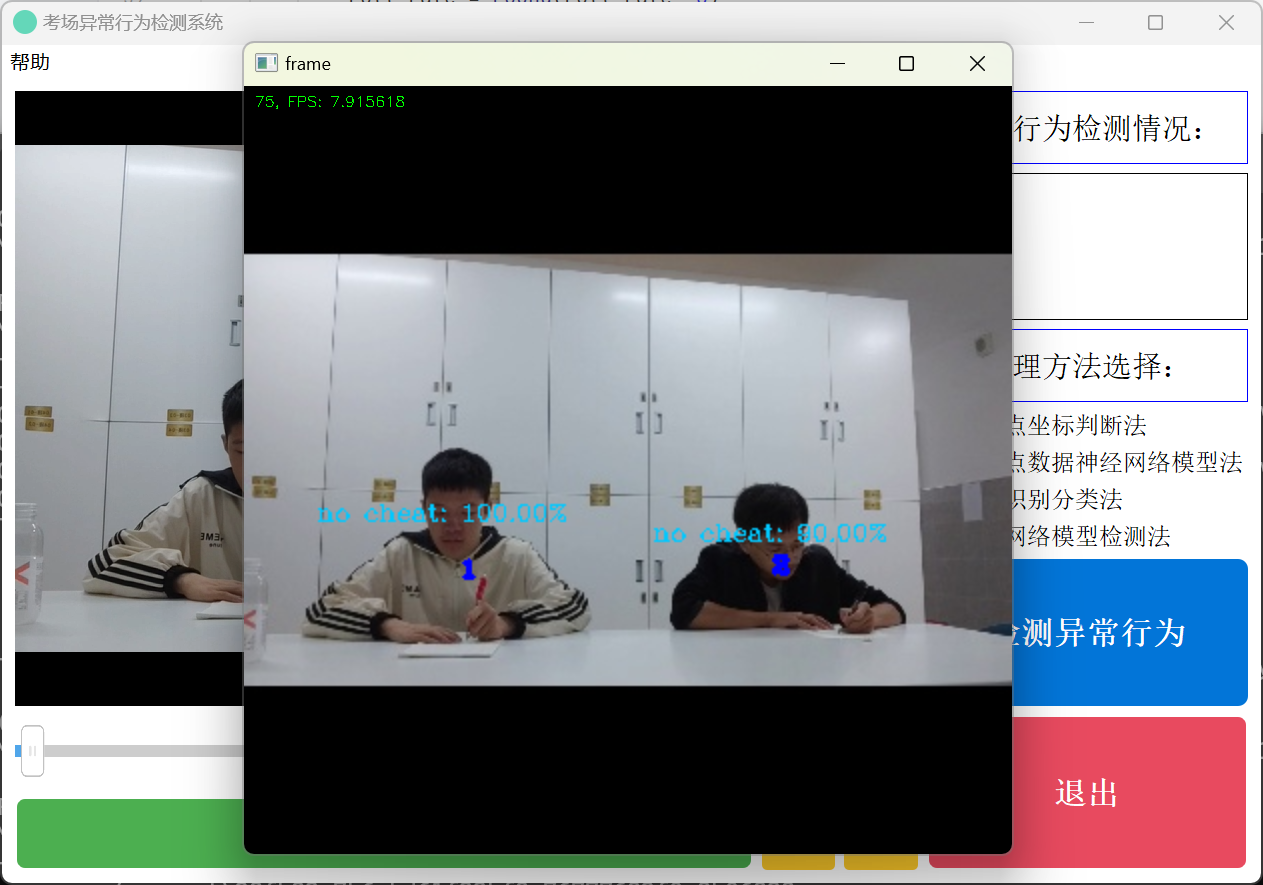


图4-15系统处理视频过程中保留了代码处理视频的每帧效果



图4-16系统处理完视频后进入“检测效果展示”新窗口

如图4-16所示，“检测效果展示”新窗口，主体布局有一个播放器，加载的是系统处理完所上传的视频之后得到的对应的效果视频，默认的是保留了骨骼点框架的视频，与主窗口的视频播放的一致，可以通过进度条查看视频播放进度，并可以拖拽进度条对视频播放进度进行控制，通过“播放”和“暂停”按钮控制视频的播放和暂停。在界面的最下方有两个“保存检测结果视频”的按钮，用户可以通过点击它们将系统处理之后的效果视频保存到本地，本系统提供了两个版本，一个是显示骨骼点框架的视频，另一个是不显示骨骼点框架的视频，用户可以根据自己需求选择进行下载。而在窗口的右半部分，上面的“异常行为检测情况”提示标签和对应下面的文本框将输出系统在处理视频过程中发现疑似异常行为的时刻点，用户可以根据这里的信息对视频进行定位和查证。下面的“当前处理的方法选择”提示标签和对应下面的四个方法的复选框（不可交互）展示了当前的视频处理效果是基于怎样的方法组合。最后是红色的“退出”按钮，当用户点击此按钮时，会弹出是否确认退出的提示弹窗，如上所述，只有当用户点击此按钮退出检测效果窗口后主窗口的“检测异常行为”才可恢复交互性，相关效果图如下图4-17、4-18、4-19、4-20、4-21所示。



图4-17检测效果展示窗口播放视频处理后的效果

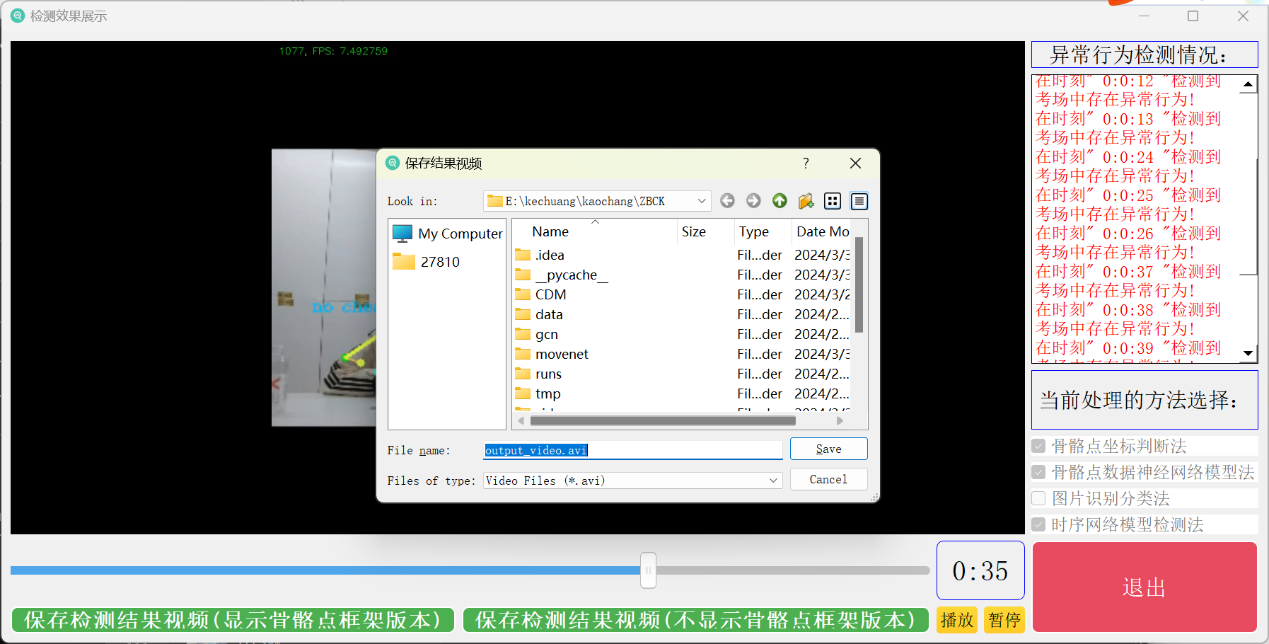


图4-18检测效果展示窗口保存处理之后的效果视频到本地

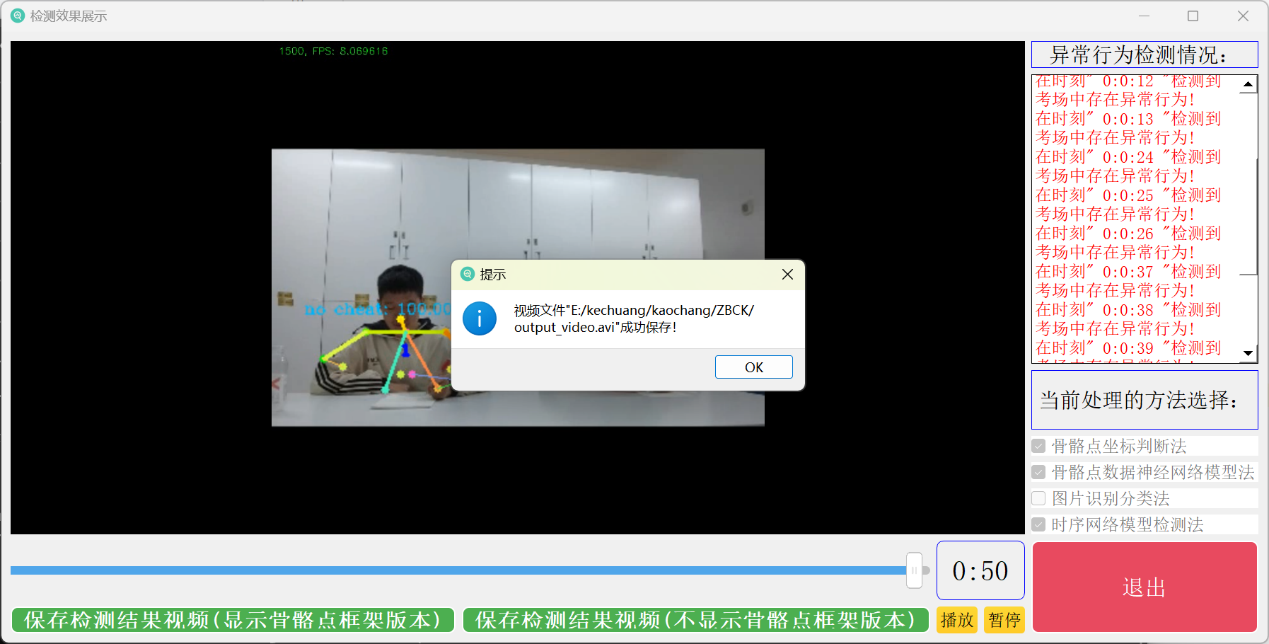


图4-19处理完成之后的视频成功保存到本地的提示

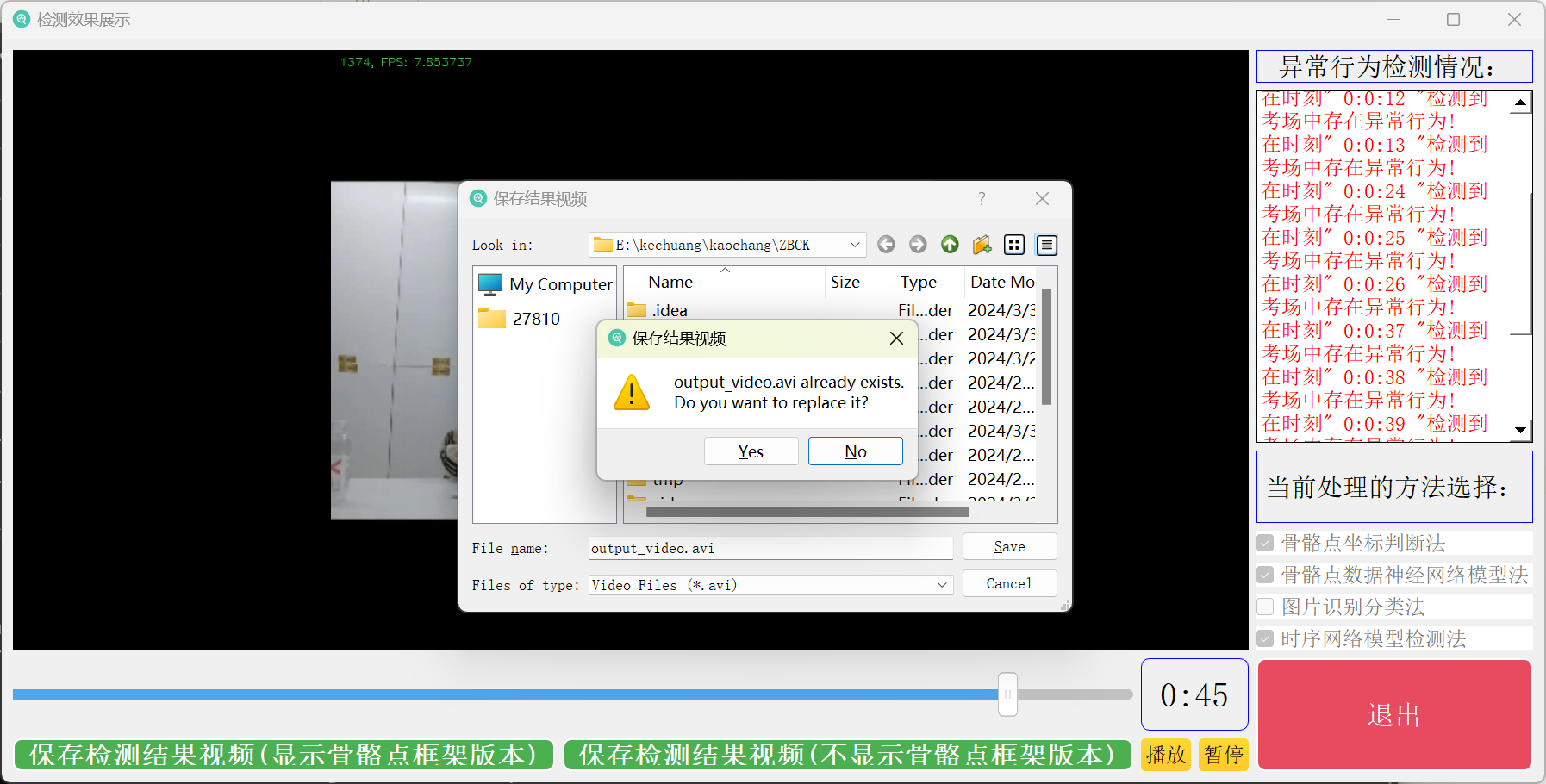


图4-20保存处理结果视频时用户选择的路径已经存在的提示

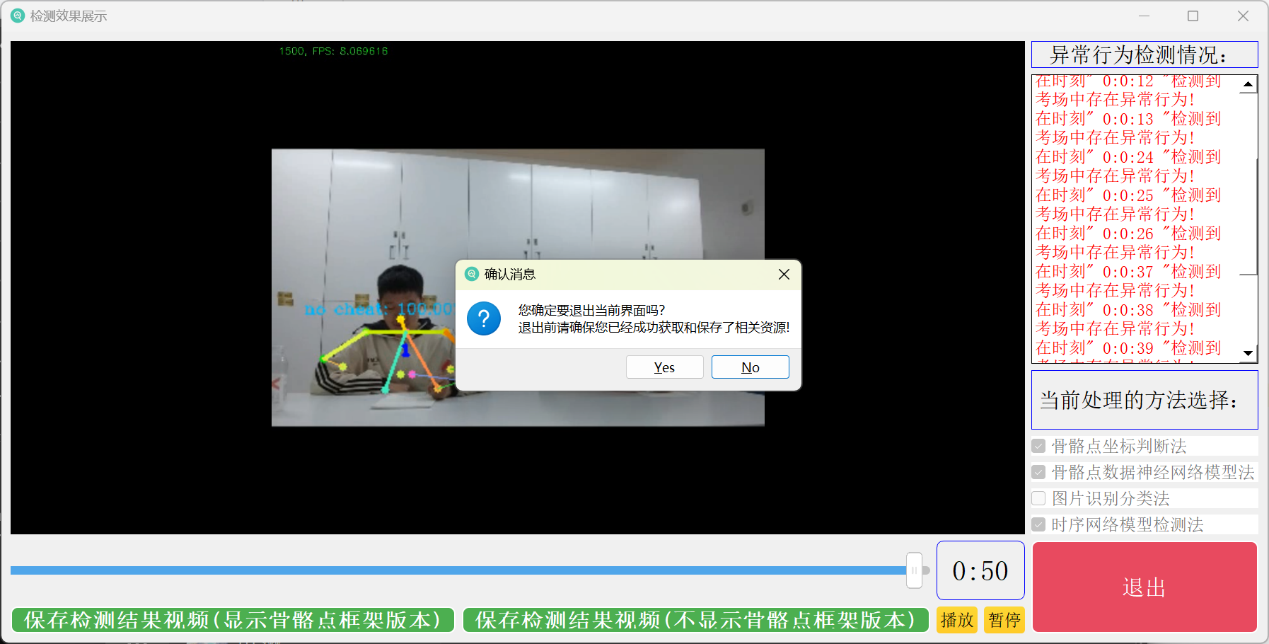


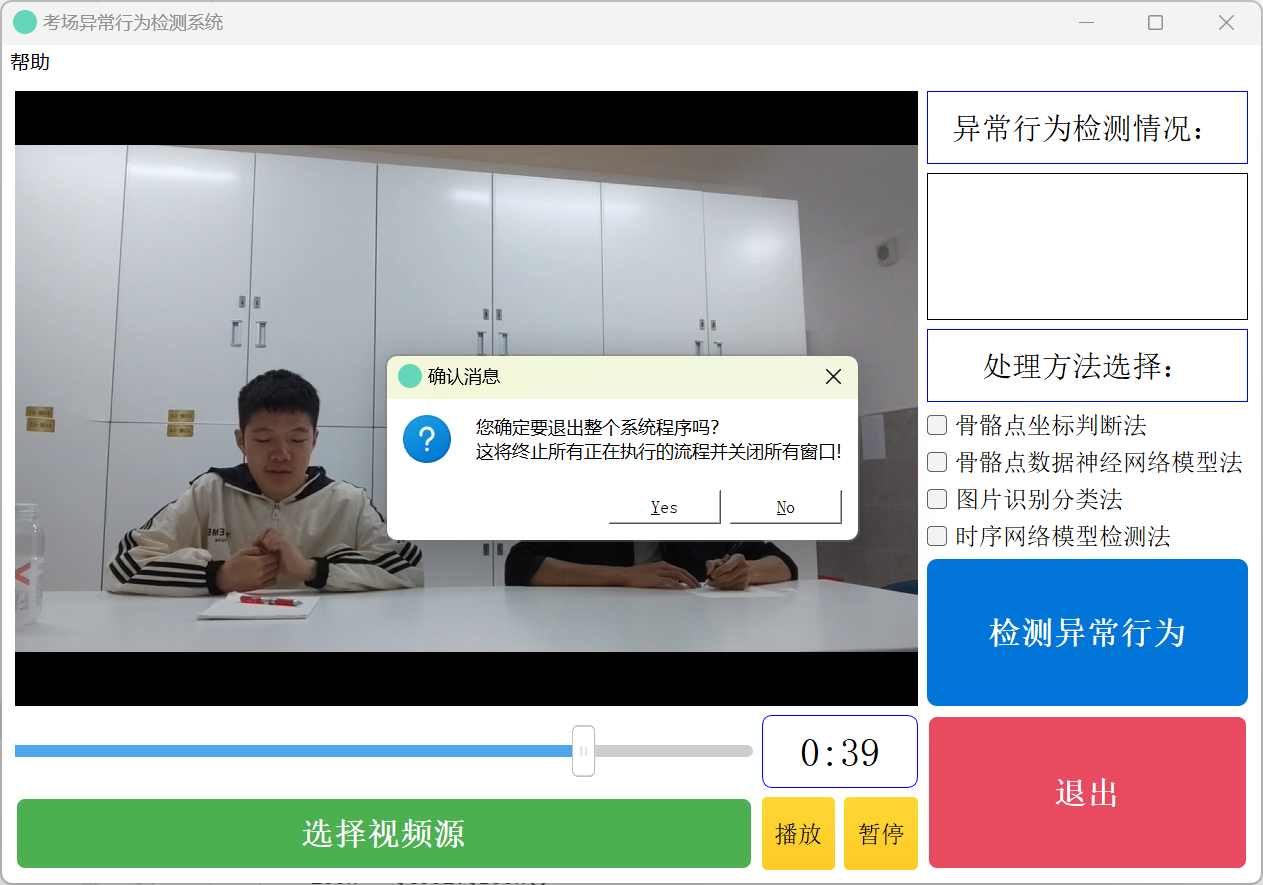
图4-21退出检测效果展示页面的提示信息

此外，考虑到用户在进行检测的过程中，可能会出现同一份视频多次处理的过程中对于同一种方法的组合方案重复进行了选取，为了避免用户在这种情况下造成的无意义的等待，对于这种情况系统将会弹出一个提示窗口并需要用户确认是否仍然选择进入检测效果展示页面，若用户仍然选择进入检测效果展示页面，由于系统在设计上单次项目的运行周期保留了所有检测的结果，因此重复进入将不会需要任何等待，用户无需担心这样的设计会使得本地磁盘的可用空间越来越少，当用户退出整个应用程序时，系统将会在后台删除所有系统程序运行周期所产生的额外文件资源，相关效果图如下图4-22所示。



图4-22用户重复处理视频的提示弹窗

最后当用户使用完本系统程序选择退出时，点击主窗口右下角的红色“退出”按钮可以退出本程序，由于是整个系统应用的退出按钮，用户点击之后系统将会弹出相关的提示，并且当用户选择确认退出之后系统程序将会中断所有正在执行的过程，并关闭所有的窗口（包括正在加载的进度条）和释放相关的资源，相关效果图如下图4-23所示。



4-23用户退出整个系统应用的提示

# 五、系统设计逻辑

声明：本部分简单介绍本系统在实现核心功能——检测异常行为的实现逻辑，与系统软件使用无关，仅提供给志同道合的研究者以供探讨。

我们使用了深度学习辅助先验知识的方法，对考场中考生的作弊行为进行检测。我们应用多种方式对视频中的单帧图像或帧序列进行处理，提取视频中有关于考生行为的当前信息与时序信息，以此识别出考生在特定时间节点是否存在疑似作弊的行为。

为了实现对视频数据的快速处理，我们应用轻量级的目标检测模型yoloV5对考场中的考生进行识别，根据所得的检测框坐标从视频的单帧图像中截取包含单名考生的图像。以单人局部图像为输入，我们应用轻量级的单人姿态估计模型MoveNet，对每名考生的骨骼点坐标进行估计。利用yoloV5所得的单人局部图像和MoveNet输出的人体骨骼点坐标这两类数据，我们分别使用基于先验知识的考场异常行为坐标判定法、基于卷积神经网络(CNN，Convolutional Neural Network)的图像分类模型、基于BP神经网络的骨骼点数据分类模型和基于时空图卷积神经网络(ST-GCN，Spatial Temporal Graph Convolutional Network)的时序分类模型，对考场中考生的疑似作弊行为进行识别。

MoveNet模型是由谷歌推出的一个轻量级人体姿态估计模型，能够输出人体17个关键点的坐标，输出骨骼点顺序如下图5-1所示。

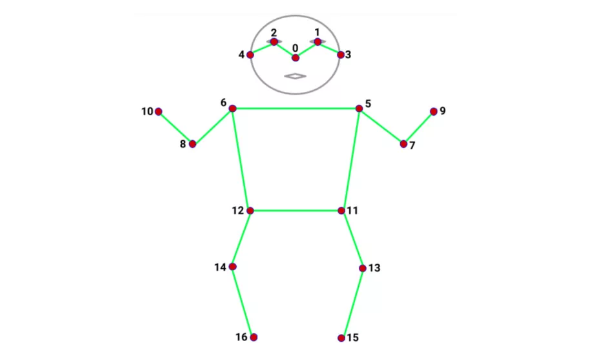


图5-1 MoveNet检测骨骼点坐标的输出信息

在基于 ResNet 的方法，考虑到处于异常状态的人与非异常状态的人在视觉上本身就具有很大的不同，尤其是对于经过 YOLO 分割出来的单人图像，背景信息的干扰已经相当少了，所以我们认为可以直接将其作为一个图像二分类问题进行处理。于是我们选择了在图像分类领域已经相当成熟的 ResNet 网络，同时为了兼顾网络预测速度，我们选择了规模较小的 ResNet34 网络。将经过 YOLO 识别然后分割出来的单人图像经过简单的预处理后，直接喂入 ResNet 网络，对于 ResNet 网络输出的 1000 维特征向量，再将其喂入一个1000\*256\*2 的全连接网络，然后再经过 softmax 层，得到预测结果。

对于基于 MoveNet + 全连接的方法，因为我们系统中考虑的异常行为都有着明显的身体动作，所以其实只要能够抽象出被识别对象的骨骼信息，就可以进行异常行为预测。因为我们已经先使用 YOLO 划分出了每个人的单人图像，所以我们选择了目前在单人骨骼点识别方面速度与精度较高的 MoveNet 网络进行骨骼点识别。同时，因为考场能捕捉到的图像大多都是上半身图像，所以对于下半身骨骼点的预测几乎是不可靠的，所以对于 MoveNet 输出的17个骨骼点信息，我们只选取了上半身的11个骨骼点进行处理。考虑到骨骼点位置与具体是否是异常动作之间的关系可能比较复杂，于是我们选择训练一个全连接网络来自动拟合这种关系。将经过 YOLO 识别然后分割出来的单人图像，先喂入 MoveNet 网络，得到人体的17个骨骼点的位置与其置信度，然后筛选出上身的11个骨骼点位置，得到一个22维的骨骼点数据，然后将骨骼点数据喂入一个 22x128x32x2 的全连接网络，再经过一个 softmax 层，得到预测结果。

在基于先验知识的考场异常行为坐标判定法中，我们利用考生出现作弊行为时其骨骼点坐标所满足的一般空间关系，从身位侧倾、转头、伸手这三方面对考生的作弊行为进行判断。身位侧倾方面，我们研究图1中由关键点0、5、6所构成的三角形，我们认为当∠065或∠056超过某一阈值(默认为90°)时，人体就发生了较大幅度的侧倾；转头方面，我们研究图1中的关键点1、2、3、4，当左耳(3)和右耳(4)分别出现在左眼(1)与右眼(2)的同侧时，我们认为考生出现了转头行为，且我们使用左耳到左眼的距离D13与右耳到右眼的距离D24的大小关系去衡量转头的幅度，当min{D13, D24}/max{D13, D24}越接近1，转头幅度就越大，考生则更有作弊的可能；伸手方面，我们研究图1中由关键点6、8、10和由关键点5、7、9构成的三角形，考生在正常答题的情况下，∠8与∠7应该为锐角，当∠8或∠7为钝角甚至接近平角时，表明考生有很大可能出现了伸手行为。但在实际检测过程中，若不是以正视角拍摄考生，则误将考生正常答题行为识别为伸手行为的可能性增加。所以，仅有当考生身位倾向一侧的手臂角度接近于平角时，我们才认为考生做出了伸手的动作。

ST-GCN是一种用于处理时空数据的图卷积神经网络。对于视频序列中的人体骨骼点，单帧视频中的骨骼点数据满足一定空间位置关系，而连续若干帧中的同一骨骼点满足一定的时序关系，因此视频序列中的骨骼点数据具有时空性质，能够作为ST-GCN模型的输入。ST-GCN模型能够对形状为(t, v, c)的骨骼点向量进行处理，并识别当前帧中的考生是否在进行作弊行为。(t, v, c)中，t表示输入的视频序列帧数，可根据实际情况进行确定，默认为10；v表示单人骨骼点数量，骨骼点种类与数量均可根据需要自行确定，这里我们对MoveNet输出的17个骨骼点数据进行处理，得到了13个骨骼点数据；c表示骨骼点数据的维数，通常为三个维度，分别表示骨骼点坐标(x,y)和置信度score。在处理单个视频时，对于由目标检测模型识别出的每一名考生，我们会使用最大容量为t的队列保存最近不超过t帧的单人骨骼点数据，当队列中存在t帧骨骼点数据时，即可作为ST-GCN模型的输入，ST-GCN模型的输出则作为当前帧的行为识别结果。

在detect\_utils.py中，我们对4种考生行为检测模型进行了整合，以ST-GCN模型的推理过程为主体，嵌入了坐标判定法、图像分类模型和骨骼点数据分类模型，实现了基于深度学习的考生异常行为的集成式检测模型。我们根据四种模型在测试集上的表现情况，将坐标判定模型、图像分类模型、骨骼点数据分类模型与ST-GCN模型的检测权重系数比例设置为0.1:0.15:0.25:0.5，通过集成模型的方式实现对考生行为更准确的预测。

# 六、合作与开源

本项目——基于智慧教室的考场异常行为检测系统是由来自南京航空航天大学（NUAA）计算机科学与技术学院的四名本科生共同完成设计与部署，本着共同探讨与进步的原则，本项目完全开源，开发组也热切希望广大仁人志士能够一同参与到本项目的改进过程中，任何有益的意见均会被采纳和予以实施。

[[1]](#footnote-1)项目开源地址：<https://github.com/xiejunxin/Smart_Classroom_Detections_of_Examing.git>。

1. Copyright ©2023-2024 NUAA Project Team Composed of Xie Junxin,Chen Yan,Tang Jiayu And Chen Qiuyu,All Rights Reserved. [↑](#footnote-ref-1)