

# Secret Flappy Bird

## 1、软件介绍

### 1.1、软件介绍

软件在 pyqt5 框架下使用 python 语言编写，结合多种算法实现了以人脸识别为基础的登陆系统以及 secret flappy bird 游戏系统。

### 1.2、环境配置

软件所需依赖库如下：

```
python>=3.8
opencv-python>=4.1.1
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.41.0
protobuf<4.21.3
tensorflow>2.11.0
keras>2.11.0
```

### 1.3、部署方法

将压缩包解压到本地后，在配置好的环境下运行 main.py 即可开始游戏

```
(base) C:\Users\17823>activate torch
(torch) C:\Users\17823>E:
(torch) E:\>cd task 6
(torch) E:\task 6>python flappy_bird/main.py
```

# 1.4、软件界面

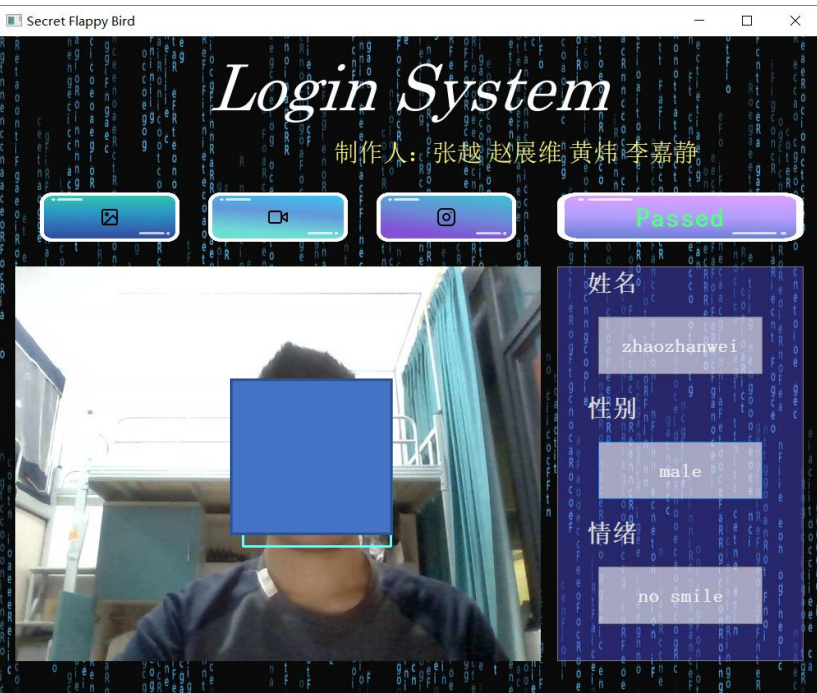
## 1.4.1 登录界面

登录界面如下，系统必须检测到特定身份的人脸后才允许我们进入游戏，检测的方式有导入照片，导入视频，开启摄像头三种，分别对应人脸显示屏上方的三个按钮。



登录界面

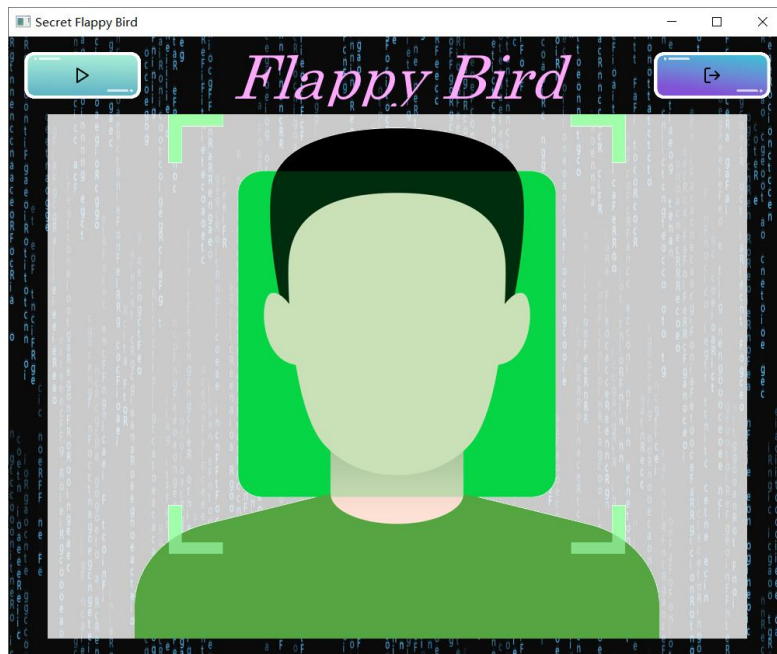
检测时，系统会在检测到的人脸中画出外接矩形框，并检测身份，性别与情绪。当身份属于特定身份时，右上角按钮会亮起，这时我们可以点击这个按钮进入游戏。



登录界面（开启摄像头）

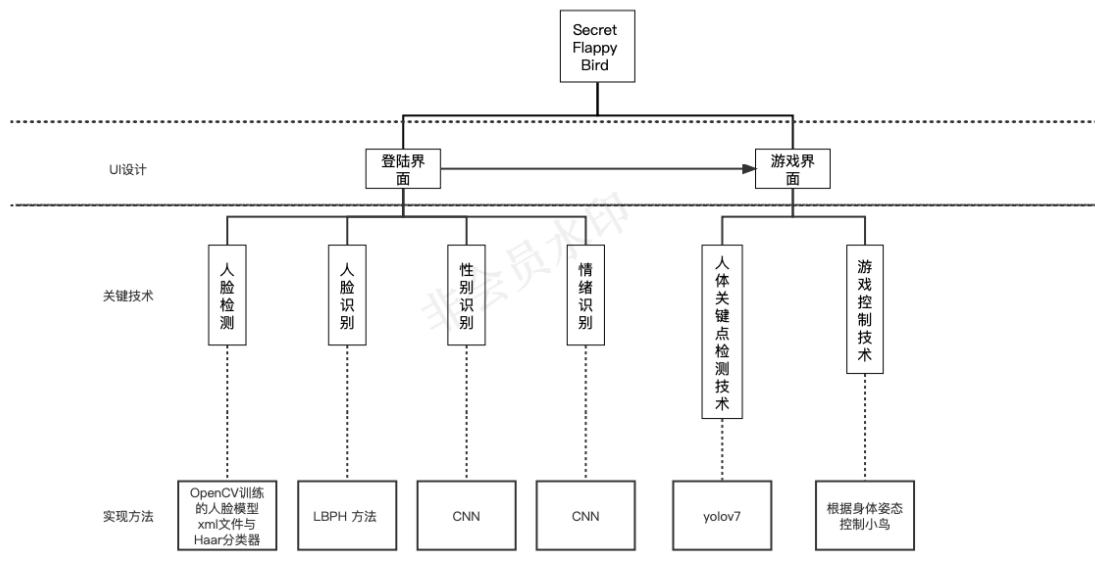
### 1.4.2 游戏界面

游戏界面如下，我们可以通过左上方的开始/暂停按钮控制游戏进程，同时可以通过右上方的退出按钮返回登录界面。



玩家通过上下摆动双臂做出“扇动翅膀”的动作，让小鸟保持在一定的水平高度，躲避画面中的“猪柱”；如果小鸟不小心碰到了“猪”或者触碰到画面边缘，游戏停止 3s，然后会重新开始。画面左上方显示的是玩家的得分，存活时间越长得分越高。

## 2、软件架构



流程图

### 3、关键技术

介绍实现的思路和关键的技术，比如人脸识别用的什么算法，对算法进行简要描述；如果有训练的，描述一下如何进行数据集的采集、训练等

#### 3.1、人脸检测技术

人脸检测通过调用 OpenCV 训练的人脸模型 xml 文件以及调用对应的 Haar 分类器，即可实现不错的检测效果。缺点是对于非正脸的检测效果不佳。我们将模型选择参数调整为：scaleFactor=1.2（默认值为 1.1），minNeighbors=5（默认为 3），这两个参数都是为了提高检测的人脸精度，保证检测的准确率。

#### 3.2、身份识别技术

##### 3.2.1 LBPH 人脸识别

人脸识别的第一步，就是要找到一个模型可以用简洁又具有差异性的方式准确反映出每个人脸的特征。识别人脸时，先将当前人脸采用与前述同样的方式提取特征，再从已有特征集中找出当前特征的最邻近样本，从而得到当前人脸的标签。

OpenCV 提供了三种人脸识别方法，分别是 LBPH 方法、EigenFishfaces 方法、Fisherfaces 方法。本项目使用 LBPH 方法。

LBPH（Local Binary Patterns Histogram，局部二值模式直方图）所使用的模型基于 LBP（Local Binary Pattern，局部二值模式）算法。LBP 最早是被作为一种有效的纹理描述算子提出的，由于在表述图像局部纹理特征上效果出众而得到广泛应用。

### 3.2.2 基本原理

LBP 算法的基本原理是，将像素点 A 的值与其最邻近的 8 个像素点的值逐一比较：

- 如果 A 的像素值大于其临近点的像素值，则得到 0。
- 如果 A 的像素值小于其临近点的像素值，则得到 1。

将像素点 A 与其周围 8 个像素点比较所得到的 0、1 值连起来，得到一个 8 位的二进制序列，将该二进制序列转换为十进制数作为点 A 的 LBP 值。

128	36	251
48	76	9
11	213	99

1	0	1
0		0
0	1	1

完成二值化以后，任意指定一个开始位置，将得到的二值结果进行序列化，组成一个 8 位的二进制数。例如，从当前像素点的正上方开始，以顺时针为序得到二进制序列“01011001”。

最后，将二进制序列“01011001”转换为所对应的十进制数“89”，作为当前中心点的像素值，如图所示。

128	36	251
48	76	9
11	213	99

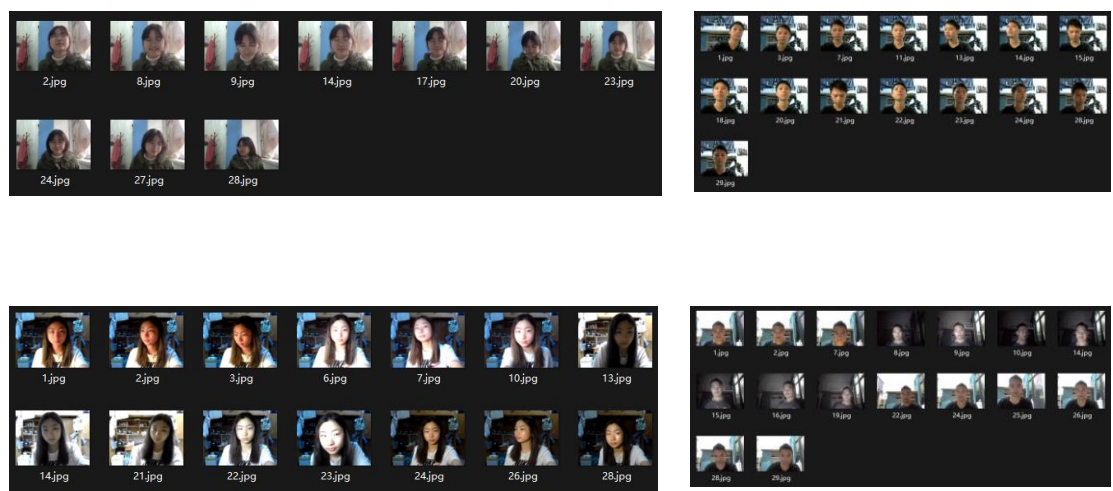
	89	

对图像逐像素用以上方式进行处理，就得到 LBP 特征图像，这个特征图像的直方图被称为 LBPH，或称为 LBP 直方图，其优点是会受到光照、缩放、旋转和平移的影响。

人脸的整体灰度由于受到光线的影响，经常会发生变化，但是人脸各部分之间的相对灰度会基本保持一致。LBP 的主要思想是以当前点与其邻域像素的相对关系作为处理结果，正是因为这一点，在图像灰度整体发生变化（单调变化）时，从 LBP 算法中提取的特征能保持不变。因此，LBP 在人脸识别中得到了广泛的应用。

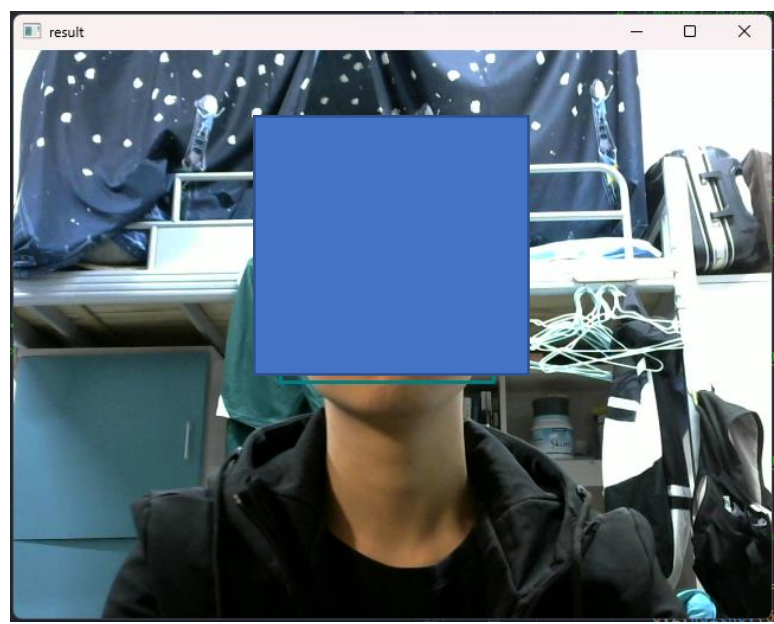
### 3.2.3 实现方法

通过采集组员在不同光照、面部角度的照片各 30 张，通过人脸检测技术得到脸部图像，舍弃无法识别人脸的无效照片，将保留下来的图像用于训练 LBPH 模型。



用于训练的图像数据集

模型识别效果



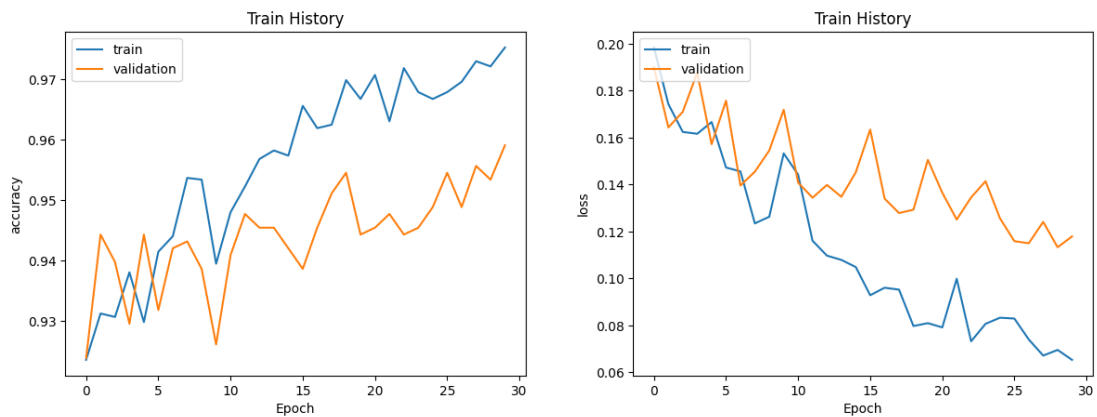
### 3.3、性别识别技术

创建一个根据人脸识别性别的 CNN，采用 SCUT-FBP5500 数据集（标记性别：男，女）进行训练。其中 3520 张图片用于训练，880 张图片用于验证，1100 张图片用于测试，训练集：验证集：测试集=8:2:2.5。测试集的准确率为 97%。

数据预处理：统一图片大小、打乱图片顺序、分割训练集与测试集、特征缩放。



训练：从训练集中划分 20%作为验证集，批大小 64，迭代 30 次，进行训练。测试集和验证集的准确率与损失随迭代次数的变化如下：



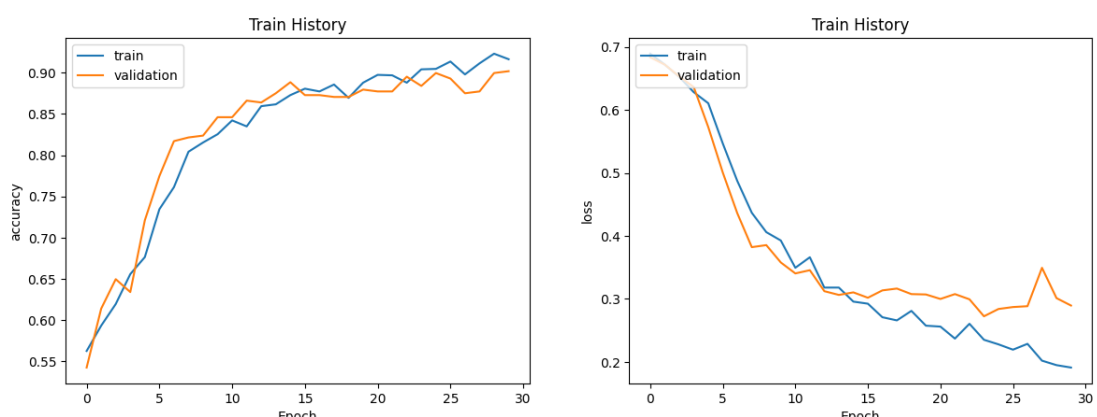
模型评估：用训练后的模型对测试集进行性别识别，准确率为 96%。

应用：利用预训练的 Haar 级联模型来检测图像中的面部，对面部用模型进行性别识别。

### 3.4、情绪识别技术

创建一个根据人脸识别情绪的 CNN，采用 GENKI 数据集（标记笑与不笑）进行训练。其中 1792 张图片用于训练，448 张图片用于验证，560 张图片用于测试，训练集：验证集：测试集=8:2:2.5。测试准确率 86%。

数据预处理与训练过程与性别检测技术一致，此处不赘述。测试集和验证集的准确率与损失随迭代次数的变化如下：



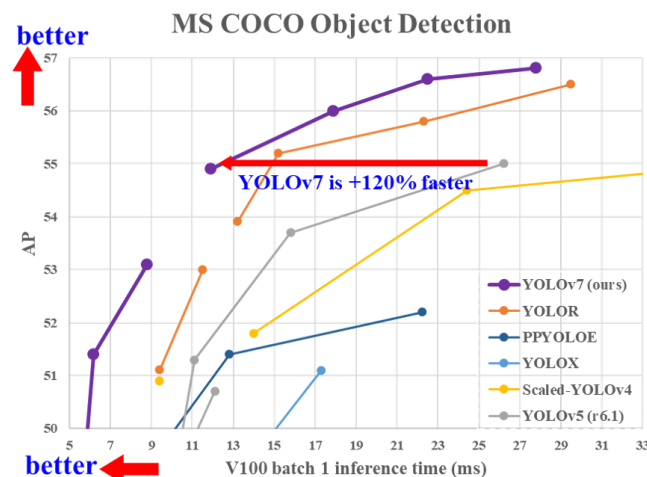
模型评估：用训练后的模型对测试集进行情绪识别，准确率为 86%。

应用：利用预训练的 Haar 级联模型来检测图像中的面部，对面部用模型进行情绪识别。

### 3.5、基于 yolov7 的人体关键点检测技术

YOLOv7 在 5 FPS 到 160 FPS 范围内，速度和精度都超过了所有已知的目标检测器，并在 GPU V100 上，30 FPS 的情况下达到实时目标检测器的最高精度 56.8% AP。YOLOv7 是在 MS COCO 数据集上从头开始训练的，不使用任何其他数据集或预训练权重。

此外，YOLOv7 的速度和精度上的表现也优于 YOLOR、YOLOX、Scaled-YOLOv4、YOLOv5、DETR 等多种目标检测器。



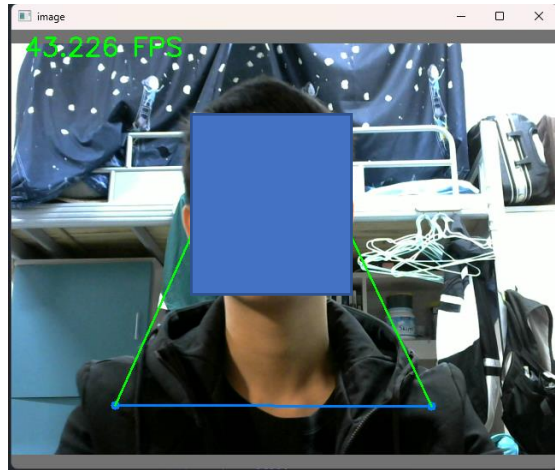
YOLOv7 的发展方向与当前主流的实时目标检测器不同，有以下特点：

- 1、能够同时支持移动 GPU 和从边缘到云端的 GPU 设备；
- 2、设计了几种可训练的 bag-of-freebies 方法，使得实时目标检测可以在不增加推理成本的情况下大大提高检测精度；我想这才是运用到工业一个巨大的提升。
- 3、对于目标检测方法的演进，研究者发现了两个新问题：一是重参数化的模块如何替换原始模块，二是动态标签分配策略如何处理分配给不同输出层的问题，并提出了解决这两个问题的方法；
- 4、提出了实时目标检测器的「扩充 (extend)」和「复合扩展 (compound scale)」方法，以有效地利用参数和计算；
- 5、该研究提出的方法可以有效减少 SOTA 实时目标检测器约 40% 的参数和 50% 的计算量，并具有更快的推理速度和更高的检测精度。

在本地部署 yolov7 的人体关键点检测模型，使用 RTX3070 显卡时的推理速度高达 40fps!

[QQ 录屏 20221217170834.mp4]

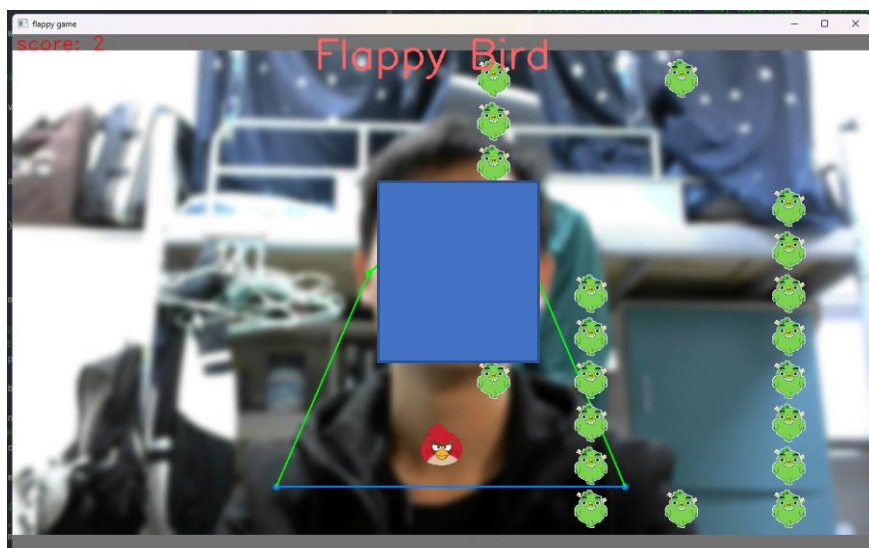


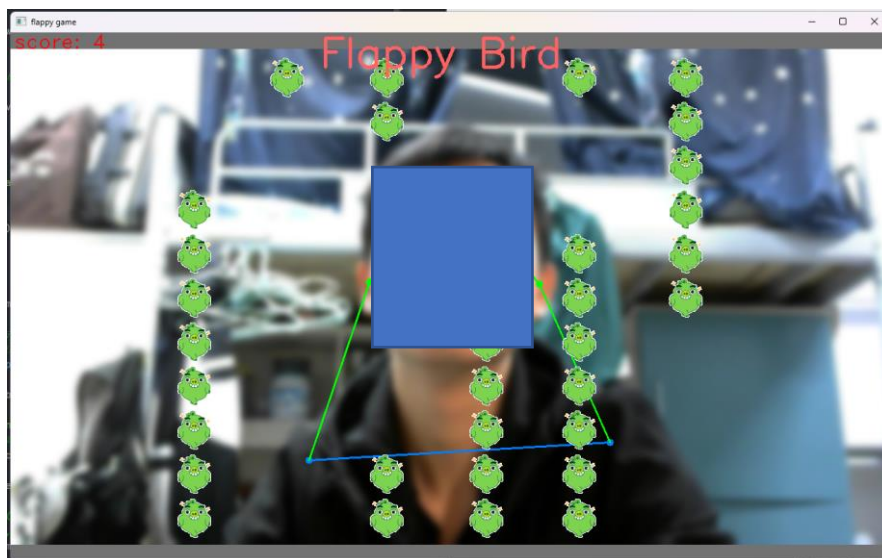


yolov7 这么快的推理速度，不用来跑游戏可惜了呀！我们经过一段时间的开发调试，将 yolov7 人体关键点检测技术和仿制的简易 Flappy Bird 游戏相结合，诞生了体感小游戏！

### 3.6、游戏控制技术

Flappy Bird 游戏与人体关键点检测应用的结合。本应用中，游戏的运行需要实时的摄像头画面，通过电脑摄像头读取画面，用 yolov7 yolov7-w6-pose.pt 预训练模型得到人体关键点数据后，计算双臂与身体的夹角，如果检测到“扇动翅膀”的动作，则控制小鸟往上飞行一段距离。游戏画面与高斯模糊后的人体关键点画面叠加在一起显示，既能看到 AI 应用的效果，也不至于干扰到游戏画面的显示。





[QQ 录屏 20221217171759.mp4]

Flappy Bird 小游戏展示了 AI 应用与游戏的结合，我们可以调整其中的游戏逻辑控制游戏的难易程度，如小鸟降落/飞升的速度、猪的出现频率与位置等；还可以改成使用其他姿态或者手势控制小鸟飞行，如跳跃、摆动身体等。

### 3.7、图形用户界面

pyqt5 是一个底层为 c++ 的 python 图形用户界面库，其运行速度快，界面美观，安全性可靠。结合 designer, pyrcc, pyuic 三个工具，可以快速的设计好图形用户界面的 UI，并将其转换为 python 代码。



同时我们还需结合多线程库 Qthreading 以及小组自行编写的算法接口，完善 GUI 的控制功能。

## 4、总结与展望

考虑到模式识别这门课的特殊性，除了学习基本的算法外，将课内学到的知识与实际应用相结合是很重要的。短短的几天时间，从选题到项目的实现，即便是奥密克戎也挡不住我们对项目的热情，虽组员不幸感染，但也坚持下来完美地完成了各项任务，令人敬佩。在完成项目的过程中，我们不仅加深了对各种算法的理解和掌握，更对于深度学习的应用、前端的制作、游戏控制等有了更深层次的认识和思考，感谢老师能给我们这样一次锻炼的机会，也感谢每一位组员的共同努力！

参考文献：

[1]<https://github.com/WongKinYiu/yolov7>

[2]<https://www.cnblogs.com/huluwa508/p/10320027.html>