

课 程 设 计 报 告

|  |  |
| --- | --- |
| **开课学期：** | 2021-2022学年第一学期 |
| **课程名称：** | 机器学习框架技术 |
| **设计题目：** | 景区安全预警系统 |
| **学 院：** | 信息科学与工程学院 |
| **专 业：** | 计算机科学与技术 |
| **团队成员：** | 郭涛 王璐鹏  司云鹏 崔昕升 |
| **班 级：** | 计智20H2 |
| **指导教师：** | 郭继昌 |

2022 年 12 月 10日

目 录

[1 深度学习背景 1](#_Toc76333886)

[1.1 深度学习历史 1](#_Toc76333887)

[1.2 深度学习的发展 1](#_Toc76333887)

[2 系统开发环境 2](#_Toc76333893)

[3 概要设计 3](#_Toc76333898)

[3.1 代码功能模块设计 3](#_Toc76333899)

[3.2 程序流程图 7](#_Toc76333900)

[4](#_Toc76333902) **[程序实现过程](#_Toc76333902)** [7](#_Toc76333902)

[4.1 模型训练与保存 7](#_Toc76333904)

[4.2 代码封装 8](#_Toc76333905)

[4.3 路由映射 8](#_Toc76333905)

[4.4 界面设计 9](#_Toc76333905)

[5 总结 1](#_Toc76333902)3

# 深度学习背景

* 1. **深度学习的历史**

一般来说，目前为止已经有三次深度学习的发展浪潮：在20 世纪 40 年代到 60 年代深度学习被称为控制论 (cybernetics)，20 世纪 80 年代到 90 年代深度学习被誉为连接机制 (connectionism)，并于 2006 年开始，以深度学习之名复兴。

连接机制是在认知科学的背景下出现的。认知科学是理解心智，并结合多个不同层次分析的跨学科方法。连接机制的中心思想是，当网络将大量简单计算单元连接在一起时可以实现智能行为。这种见解同样适用于与计算模型中隐藏单元作用类似的生物神经系统中的神经元。\n\n 神经网络研究的第三次浪潮始于 2006 年的突破。Geoffrey Hinton表明名为深度信念网络（DBN）的神经网络可以使用一种称为贪心逐层训练的策略进行有效地训练(Hinton et al., 2006a)。

* 1. **深度学习的发展**

与日俱增的数据量和模型规模,数据量的增加，使得深度学习的一些算法获取良好性能的技巧逐渐减少，目前，在复杂的任务中能达到与人类表现可以媲美的性能。大数据时代的到来，使得深度学习变得更加容易。但我们应该注意在无监督和半监督学习中充分利用未标注的样本。

数据量的增加,模型规模的扩大，较大的网络能够在更复杂的任务中实现更高的精度。所以更多学者在神经网络中引入更多的隐藏单元，使得模型规模扩大。另外，硬件（更快的CPU、通用GPU、硬盘存储量）以及更好的分布式计算的软件基础设备和更快的网络连接使得模型规模扩大成为可能。

与日俱增的精度、复杂度和对现实世界的冲击,深度学习提供精确识别和预测的能力一直在提高。此外，深度学习持续成功地应用于越来越广泛的应用。

深度网络在ImageNet挑战中日益降低的错误率,深度学习是机器学习的一种方法，过去几十年的发展中，它深深地吸收了我们关于人脑、统计学与应用数学的知识。近年来，深度学习的普及性和实用性有了极大的发展，这在很大程度上得益于更强大的计算机、更大的数据集和能够训练更深网络的技术。未来几年充满了进一步提高深度学习并将它带到新领域的挑战和机遇。

# 系统开发环境

操作系统：macOS 13.0.1 (22A400)

软件： PyCharm 2022.2.2 (Professional Edition）

编译环境： python3.8

# 概要设计

## 代码功能模块设计

**后端**：通过yolov5算法进行目标识别与分类

yolov5网络架构

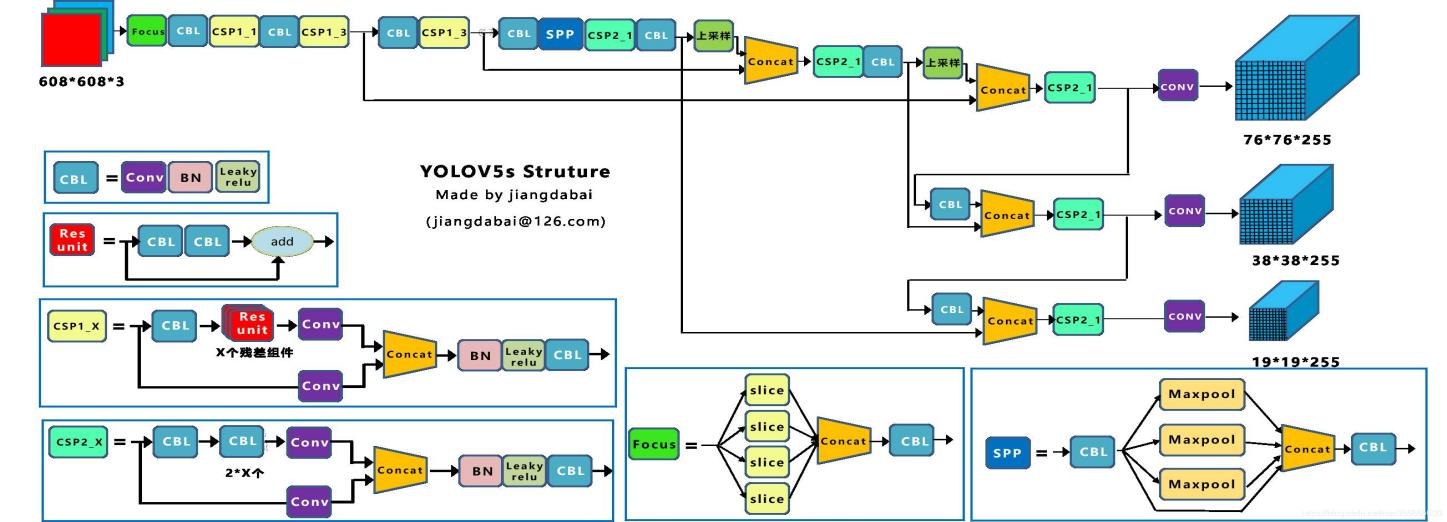


图3.1

1. **输入端**
2. Mosaic数据增强

Yolov5的输入端采用了Mosaic数据增强的方式，通过随机缩放、随机裁剪、随机排布的方式进行拼接，对于小目标的检测有很好的效果。

1. 自适应锚框计算

在Yolo算法中，针对不同的数据集，都会有初始设定长宽的锚框，在网络训练中，网络在初始锚框的基础上输出预测框，进而和真实框groundtruth进行比对，计算两者差距，再反向更新，迭代网络参数。

1. 自适应图片缩放

在常用的目标检测算法中，不同的图片长宽都不相同，因此常用的方式是将原始图片统一缩放到一个标准尺寸，再送入检测网络中。

1. **Backbone**
2. Focus结构

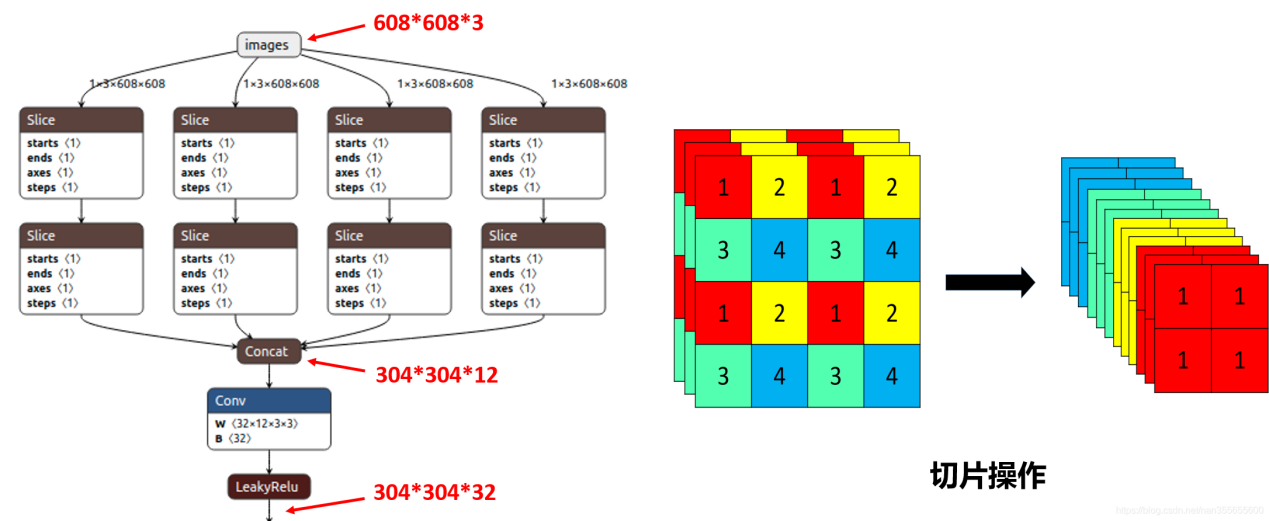


图3.2

1. CSP结构

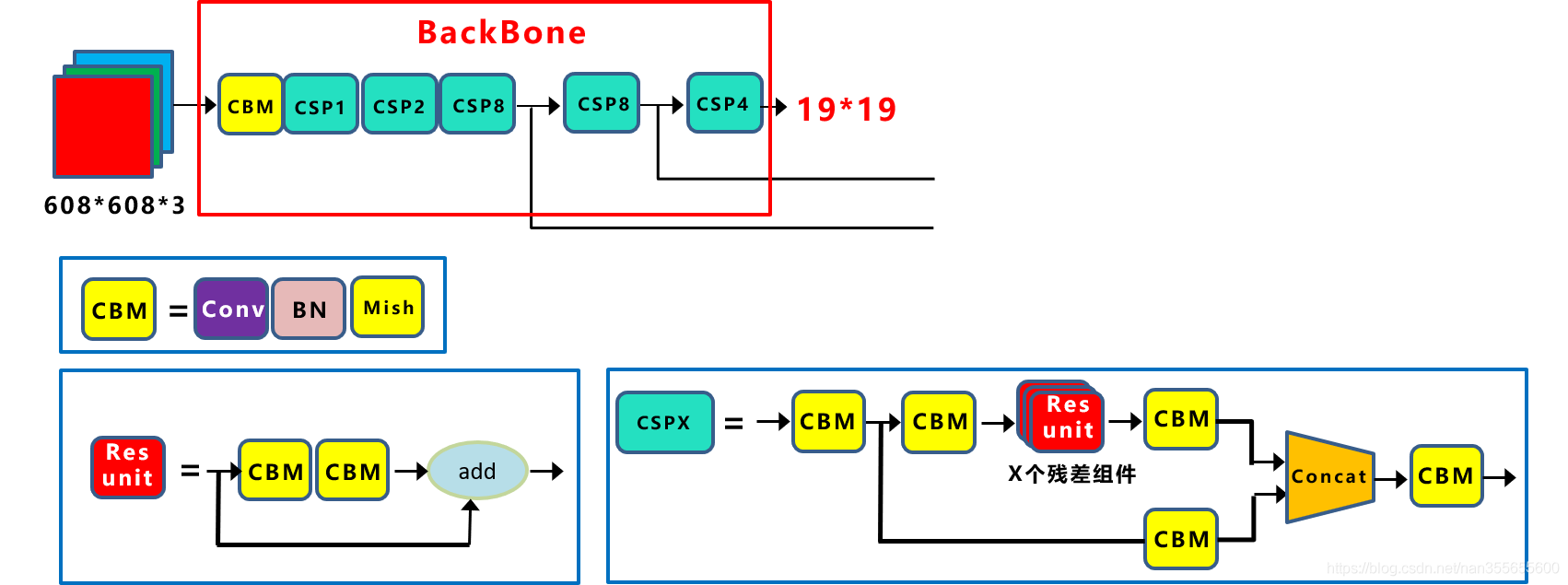


图3.3

1. **Neck**

Yolov5的Neck采用FPN+PAN的结构

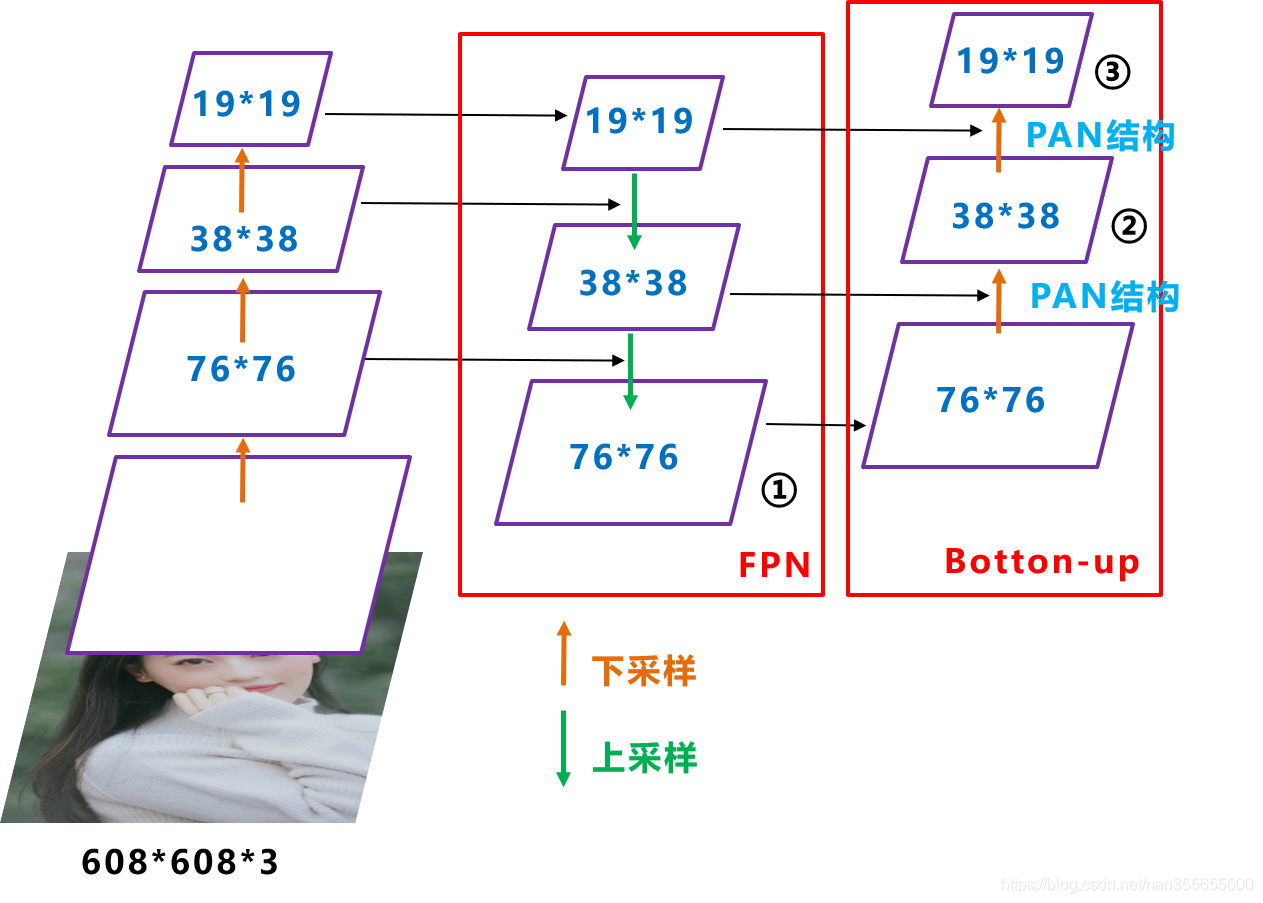


图3.4

1. **输出端**
2. Bounding box损失函数

Yolov5中采用其中的CIOU\_Loss做Bounding box的损失函数

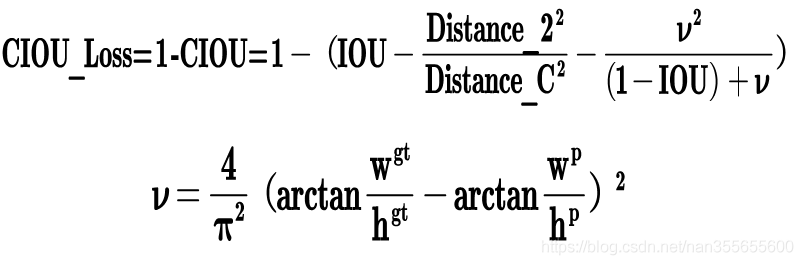


图3.5

1. nms非极大值抑制

在目标检测的后处理过程中，针对很多目标框的筛选，通常需要nms操作，Yolov5中采用加权nms的方式。

**前端**：通过Django作为服务器并进行路由映射

管理者：通过Pycharm启动Django作为服务器并启动路由映射

客户端：通过输入相应的ip地址与端口号进入index首页

界面首页

图3.2

## 程序流程图



图3.3

# 程序实现过程

## 模型的训练与保存

通过train.py进行模型训练并获得最佳权重参数，选取yolov5s.py作为初始权重参数

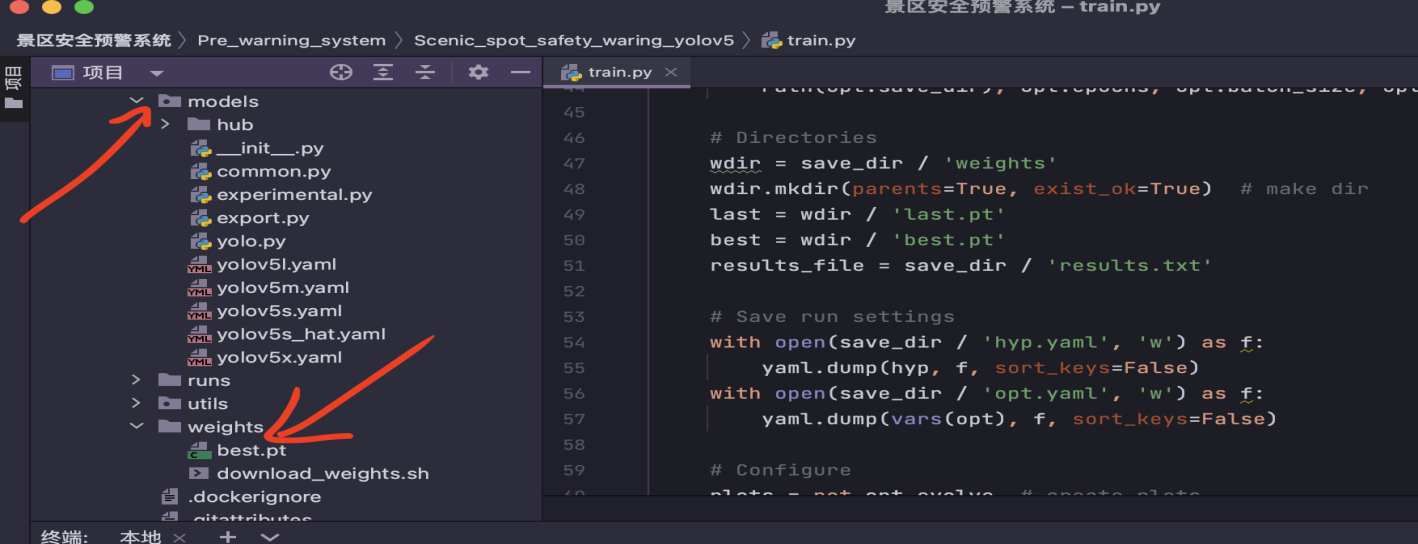
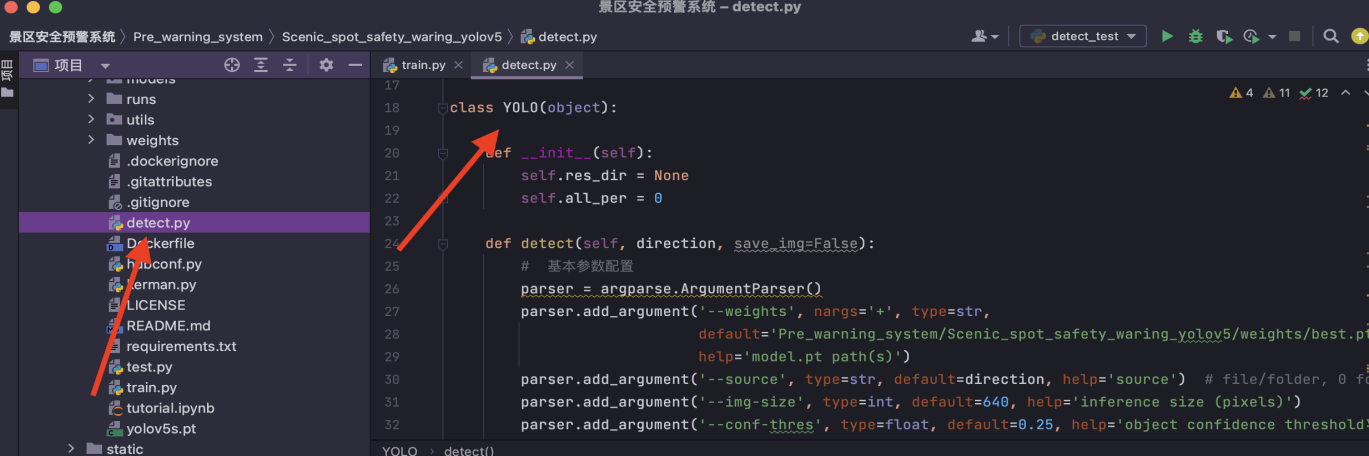


图4.1

## 代码封装

对detect.py中推理函数进行封装，封装为Yolo类便于调用

图4.2

## 路由映射

通过Django进行路由映射，分为index首页，result图片识别结果页面，video视频流，get实时获取总人数

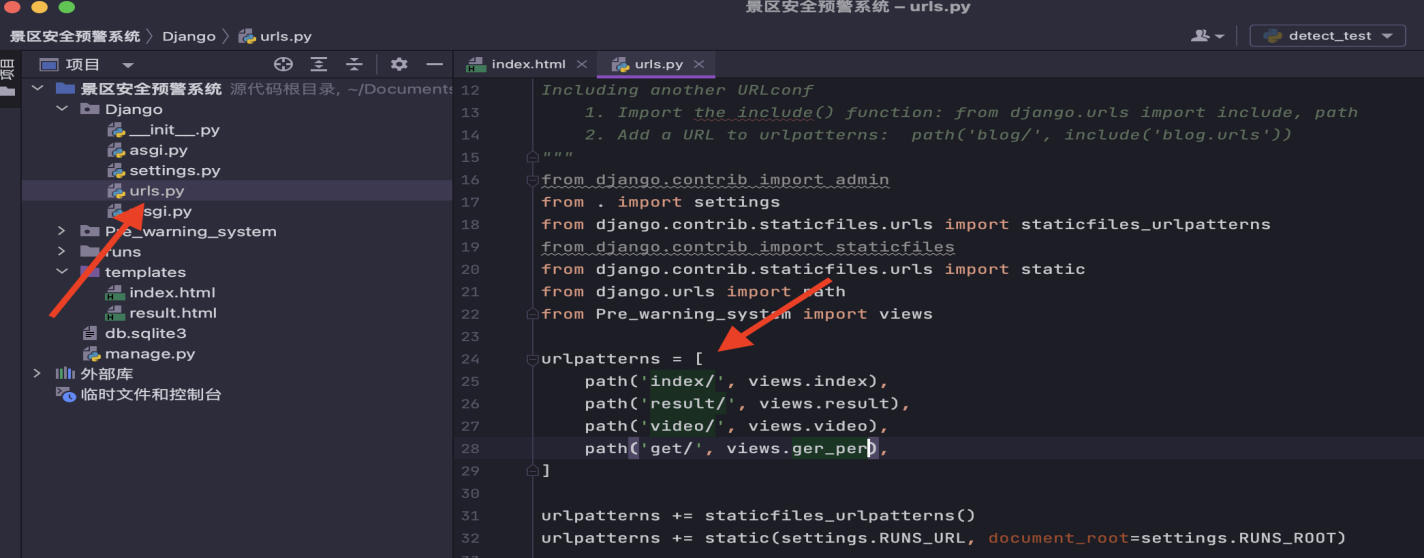


图4.3

## 界面设计

利用Bootstrap与JQuery框架进行前端界面的设计

通过点击开启摄像，调用摄像头并进行目标检测，统计人数



图4.4

调用摄像头，进行目标检测，给出相应的置信度，进行非极大值抑制获取最合适的Bounding box

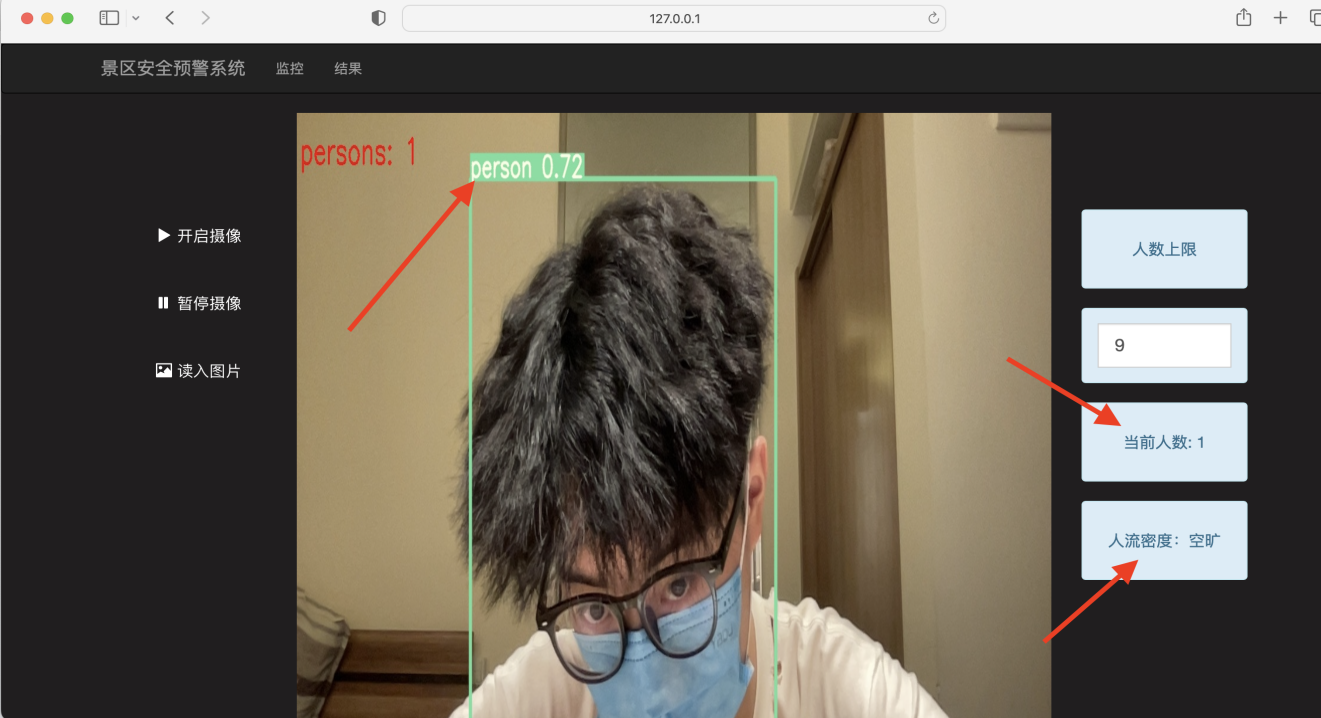


图4.5

监控界面初始人数设置为9，可通过输入框修改人数上限，人数若低于人数上限的百分之八十，人流密度为空旷，若高于人数上限的百分之八十，人流密度为拥挤，若超过人数上限，人流密度为堵塞并弹出警告框提醒

前端代码实现

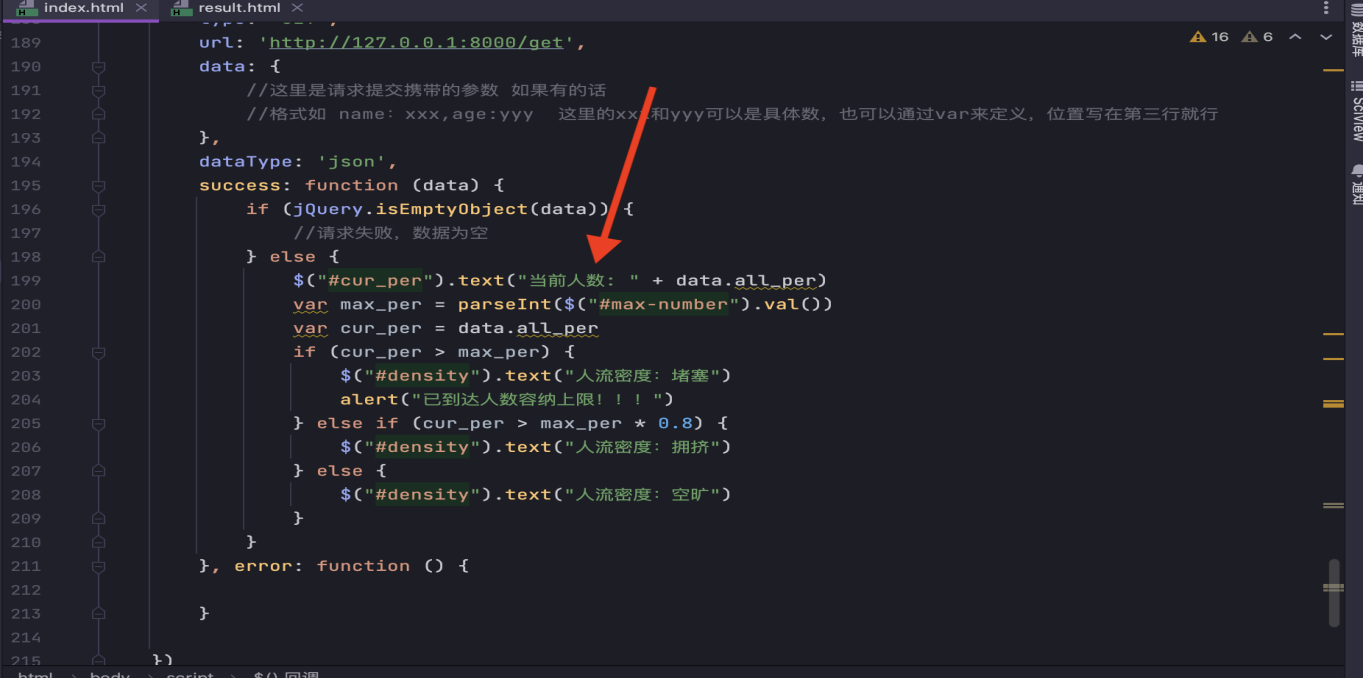


图4.6

预警提醒

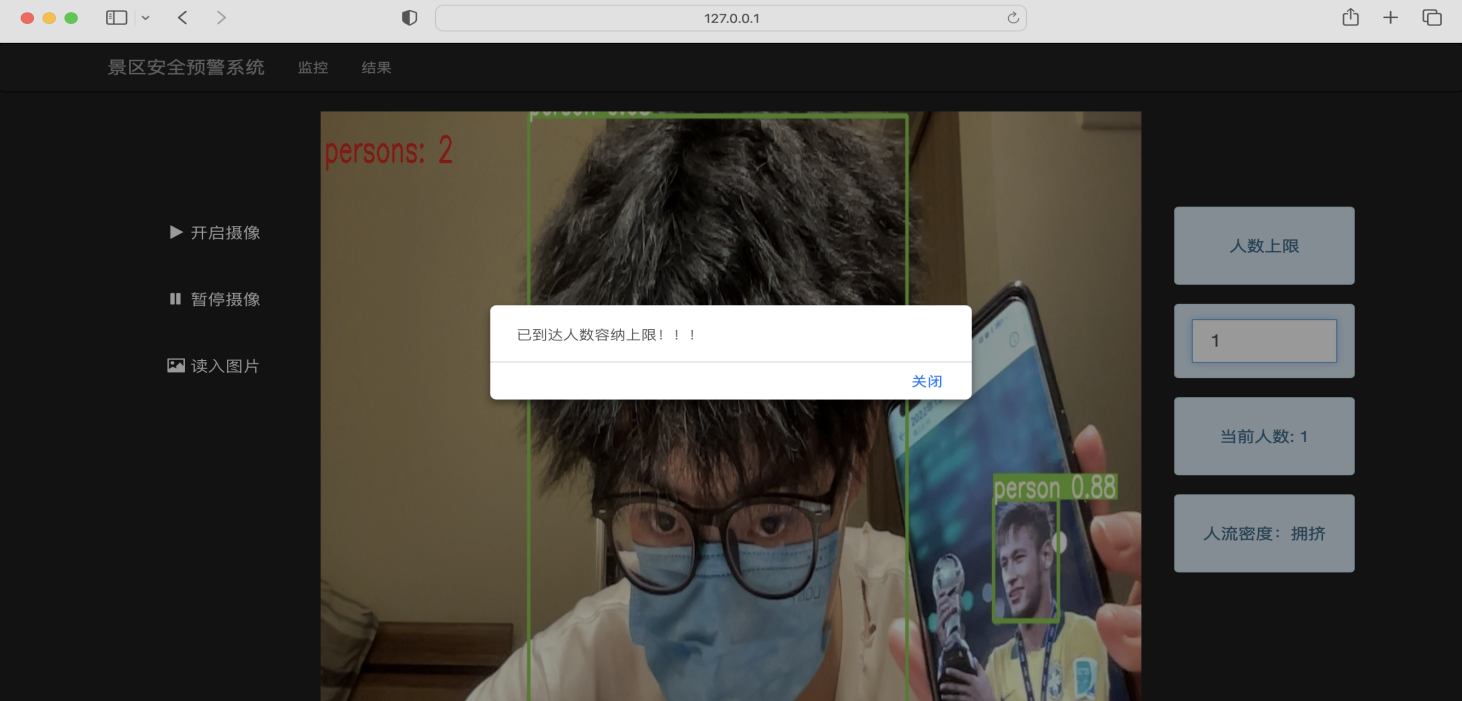


图4.7

通过点击暂停摄像关闭摄像头



图4.8

点击读入图片，可输入图片进行人数检测

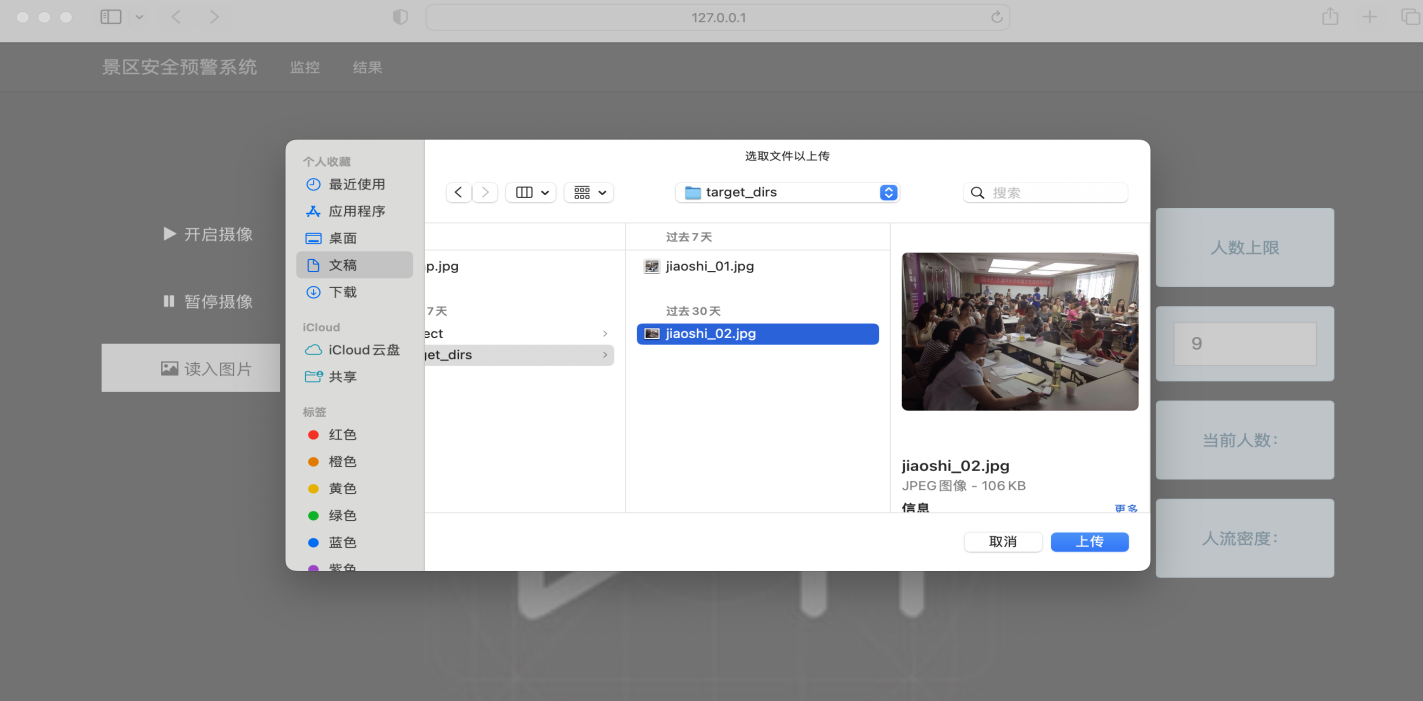


图4.9

识别结果

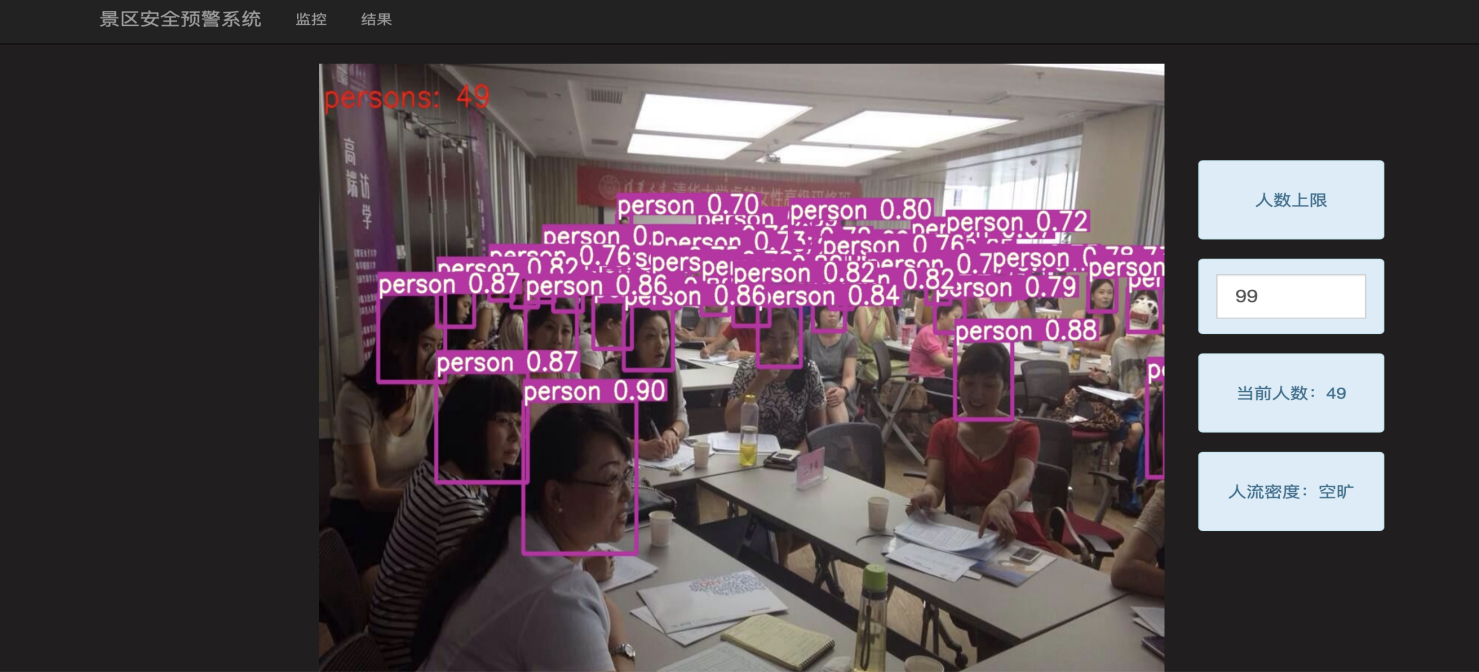


图4.10

## **5总结**

回顾起此课程设计，感慨颇多，从理论到实践，在这学期的学到很多知识，学到很多很多的东西，不仅仅巩固了以前所学过的知识，也学到很多在书本上所没有学到过的知识。在实验操作与设计的过程中遇到问题也颇多，但可喜的是最终都得到了解决。

此次课程实验学习给自我的感触是，不管什么样的算法，懂的也好不懂的也好，都要动手去用，仅有自我操作了，才会真正明白其中的用处，其次是，在遇到困难的时候，不要总是一个人闭门造车，团队间应当互相的帮忙，才能更好地克服困难。