



基于yolo框架的口罩类型及合理佩戴的 识别检测系统开发



答辩人：袁帅祥
指导老师：倪彤光老师



目录

1 研究意义

2 目标检测算法的研究

3 系统开发流程

4 总结与展望



PART ONE

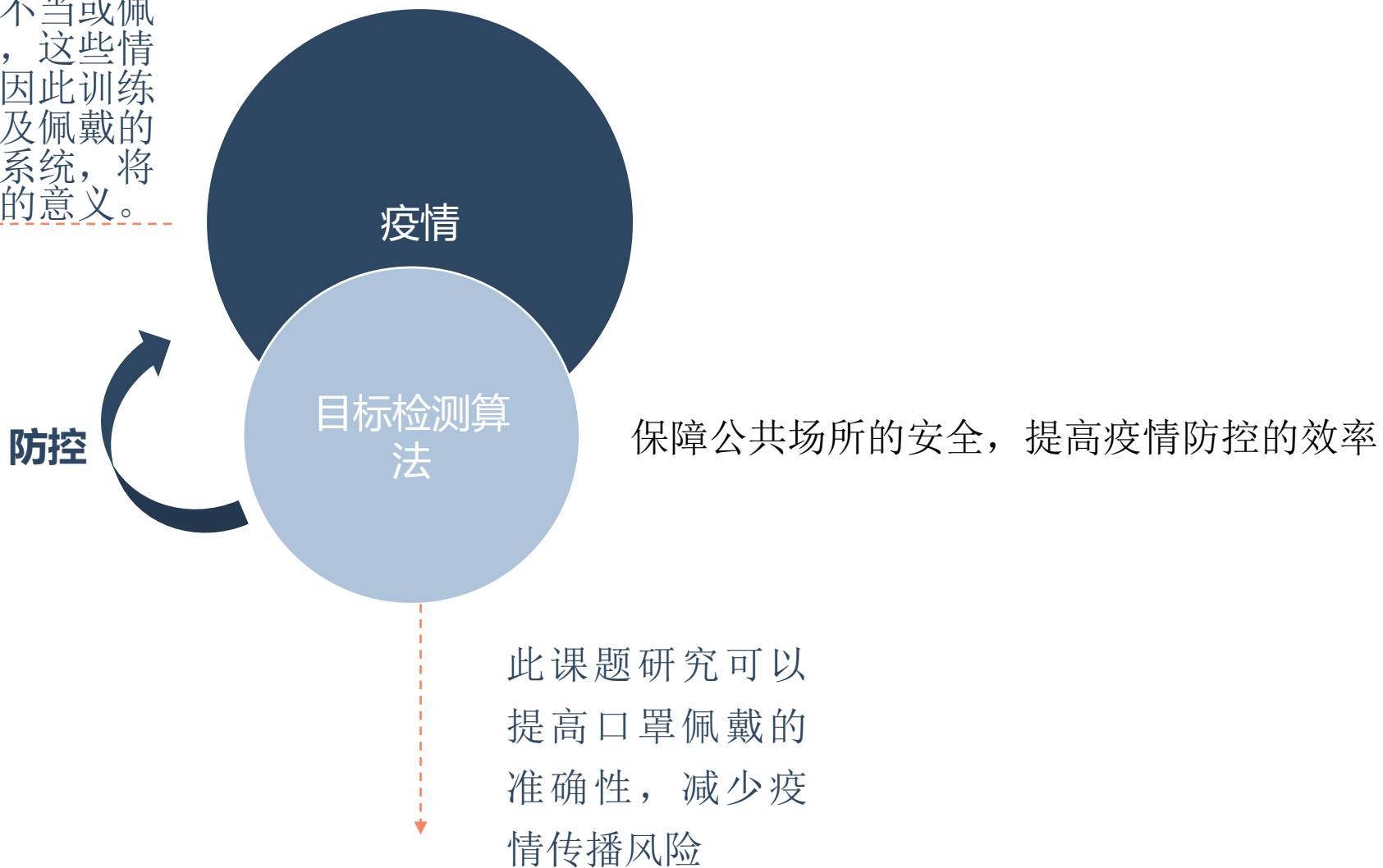


研究意义

PART ONE



随着新冠疫情的全球爆发，口罩成为了人们防疫的重要利器。然而，由于人员众多，口罩佩戴不当或佩戴不合适的情况时有发生，这些情况会增加疫情传播风险，因此训练一个口罩是否合理佩戴以及佩戴的口罩类型是否安全的识别系统，将对疫情防控工作产生重要的意义。



PART TWO



目标检测算法的研究

PART TWO



1 YOLOv5

选择了yolov5s模型，小型高效

2 RetinaNet

采用Focal loss解决了样本不均衡的问题

3 SSD

直接对不同层次和分辨率的特征图进行检测，计算速度快

4 CenterNet

anchor-free，为像素点预测锚框，预测中心点和宽高来找到目标

5 FasterRCNN

先生成候选区域，再对区域进行分类和定位，准确率更高，但是检测速度慢。

PART THREE



系统开发流程

PART THREE

数据预处理



dataset > YOLO_Mask > score > images > train

在 train 中搜索

名称	日期	类型	大小	标记
correct (481).jpg	2020/8/14 6:14	JPEG 图像	279 KB	
correct (482).jpg	2020/8/14 6:15	JPEG 图像	292 KB	
correct (483).jpg	2020/8/14 6:15	JPEG 图像	260 KB	
correct (484).jpg	2020/8/14 6:15	JPEG 图像	331 KB	
correct (485).jpg	2020/8/14 6:14	JPEG 图像	298 KB	
correct (486).jpg	2020/8/14 6:14	JPEG 图像	247 KB	
correct (487).jpg	2020/8/14 6:14	JPEG 图像	253 KB	
correct (488).jpg	2020/8/14 6:17	JPEG 图像	347 KB	
correct (489).jpg	2020/8/14 6:15	JPEG 图像	268 KB	
correct (490).jpg	2020/8/14 6:16	JPEG 图像	274 KB	

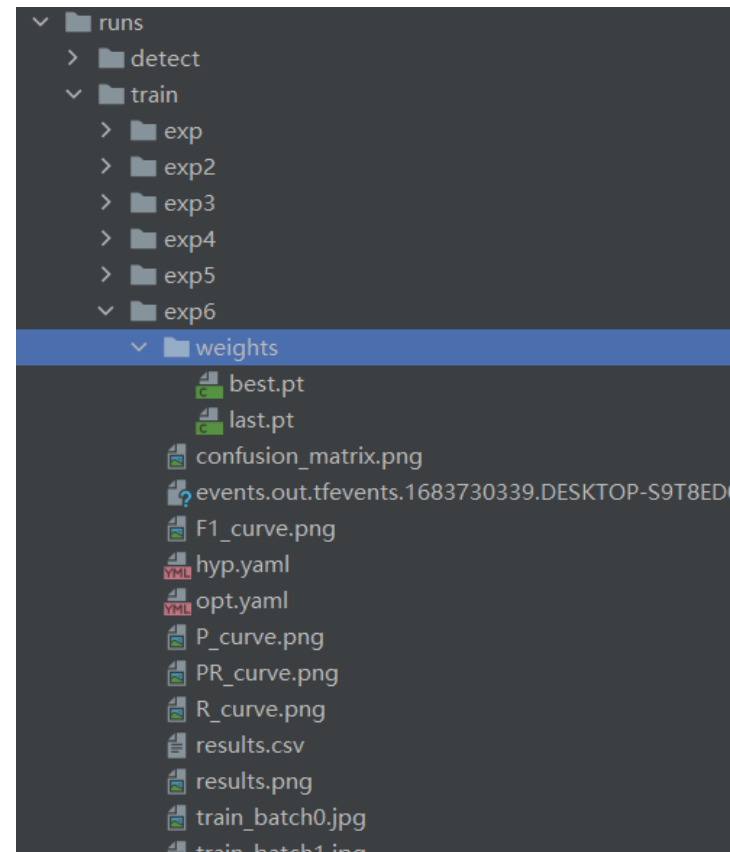
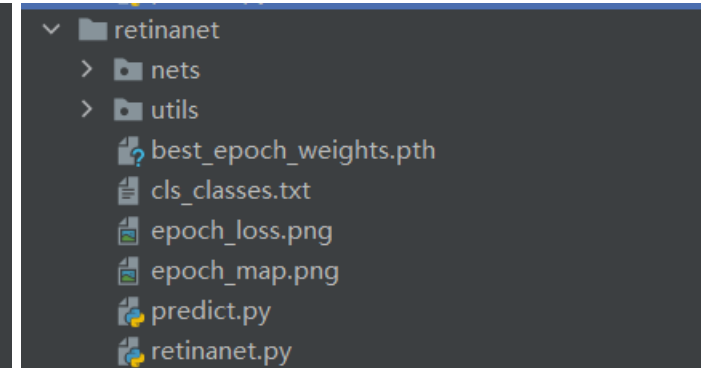
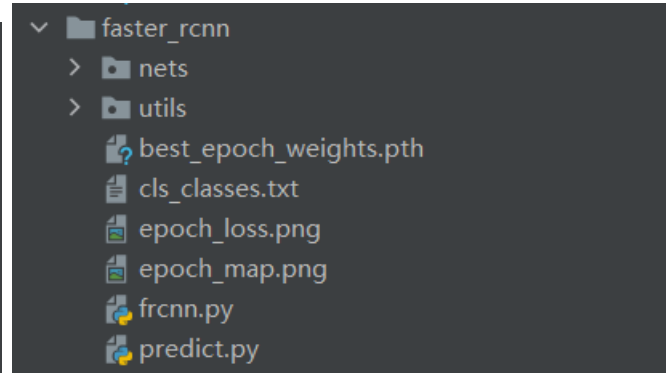
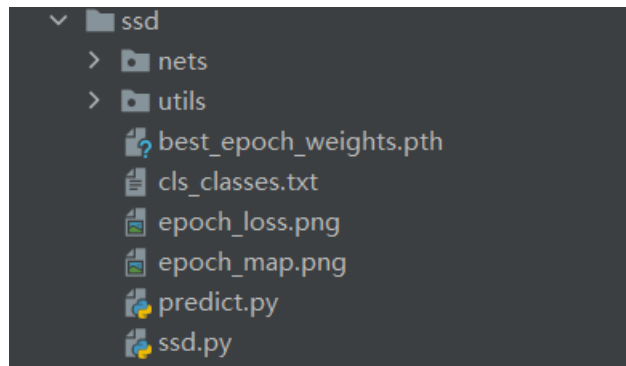
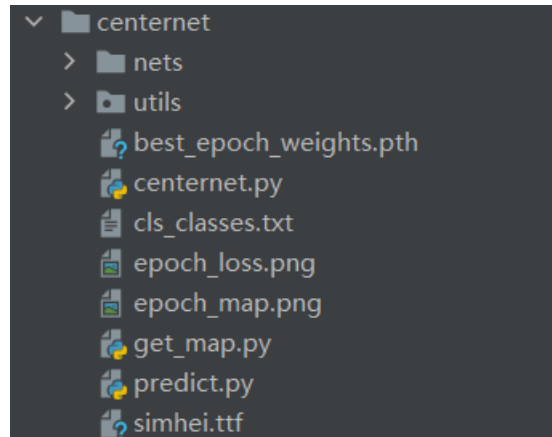
```
def makexml(picPath, txtPath, xmlPath): # txt所在文件夹路径, xml文件保存路径, 图片所在文件夹路径
    """此函数用于将yolo格式txt标注文件转换为voc格式xml标注文件
    在自己的标注图片文件夹下建三个子文件夹, 分别命名为picture、txt、xml'normal', 'correct', 'safe', 'wrong', 'face'"""
    dic = {'0': "normal", # 创建字典用来对类型进行转换
           '1': "correct", # 此处的字典要与自己的classes.txt文件中的类对应, 且顺序要一致
           '2': "safe",
           '3': "wrong",
           '4': "face",
           }
    files = os.listdir(txtPath)
    for i, name in enumerate(files):
        xmlBuilder = Document()
        annotation = xmlBuilder.createElement("annotation") # 创建annotation标签
        xmlBuilder.appendChild(annotation)
        txtFile = open(txtPath + name)
        txtList = txtFile.readlines()
        img = cv2.imread(picPath + name[0:-4] + ".jpg")
        Pheight, Pwidth, Pdepth = img.shape

        folder = xmlBuilder.createElement("folder") # folder标签
        foldercontent = xmlBuilder.createTextNode("driving_annotation_dataset")
        folder.appendChild(foldercontent)
        annotation.appendChild(folder) # folder标签结束

        filename = xmlBuilder.createElement("filename") # filename标签
```



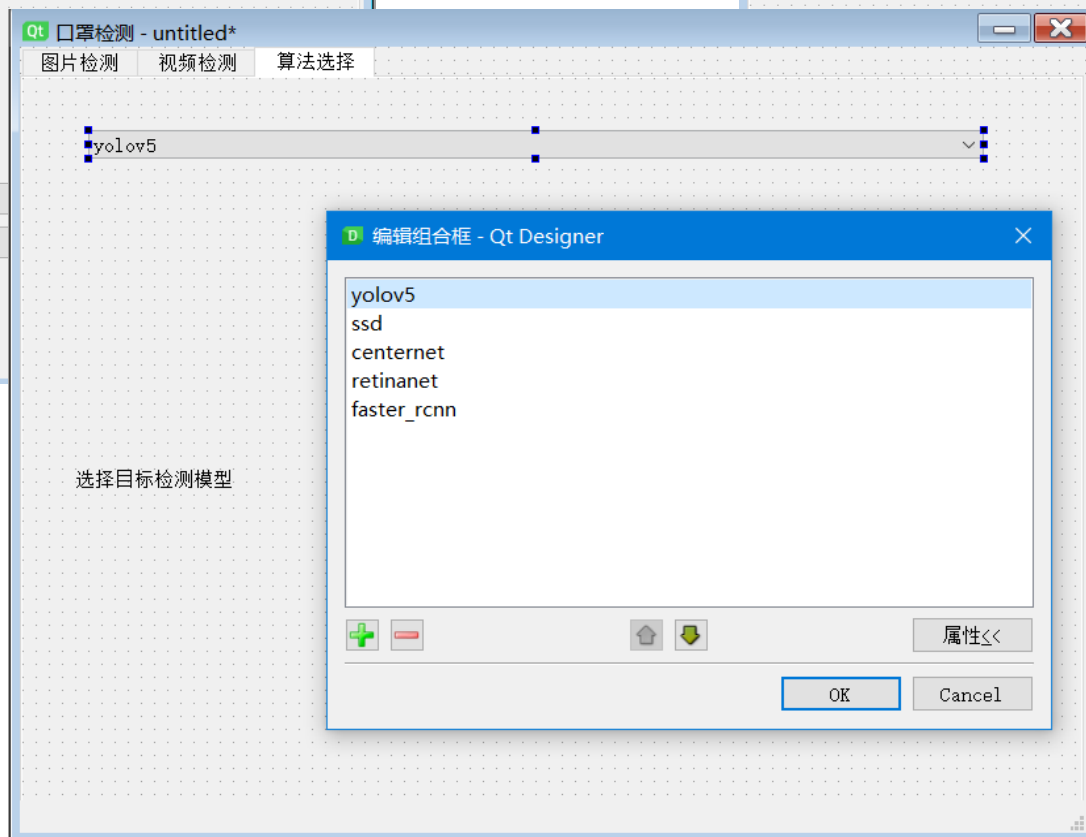
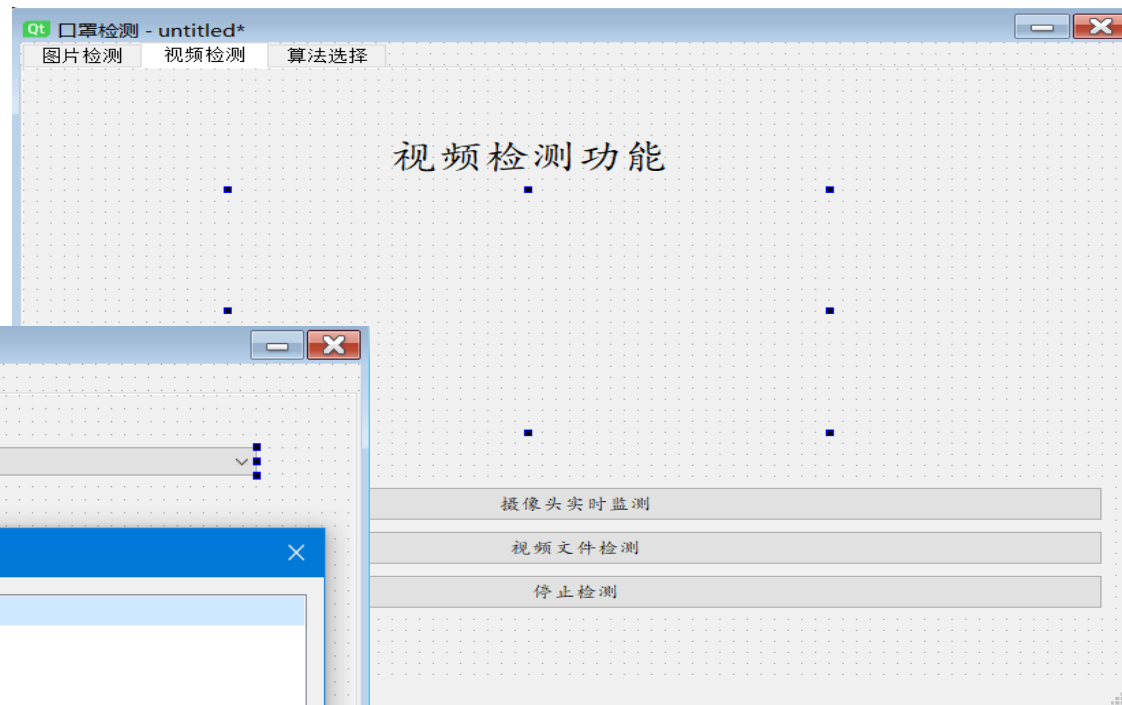
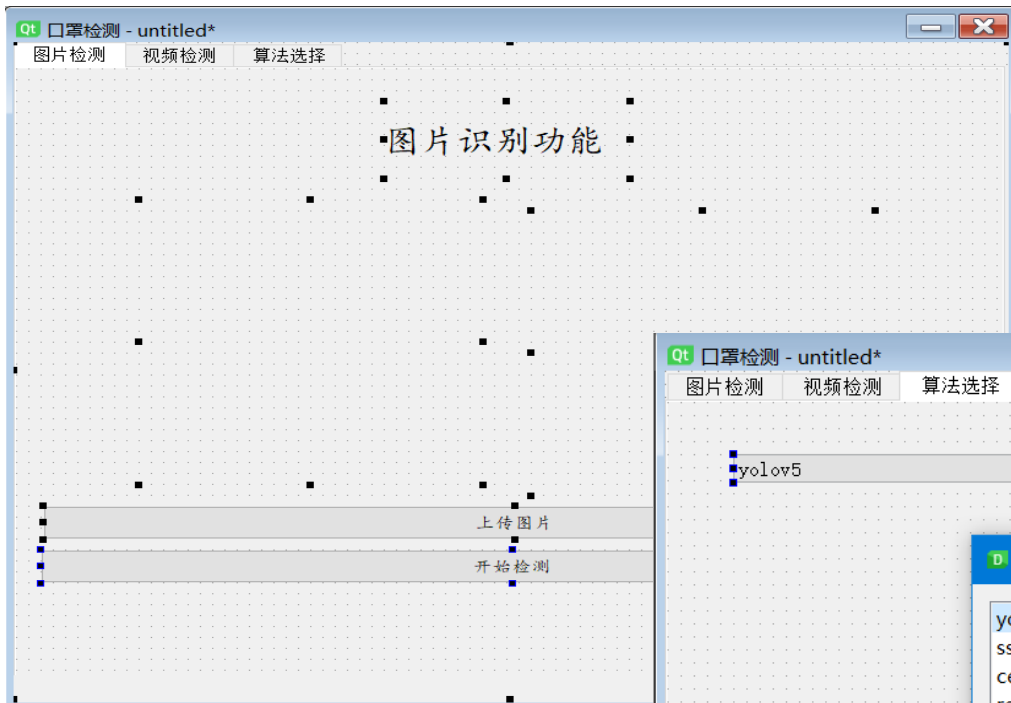
PART THREE



PART THREE



使用QtDesigner进行界面设计



PART THREE



将五种模型集成到系统中

```
fileName, fileType = QFileDialog.getOpenFileName(self, 'Choose file', '', '*.mp4 *.avi')
if fileName:
    self.webcam_detection_btn.setEnabled(False)
    self.mp4_detection_btn.setEnabled(False)
    self.vid_source = fileName
    self.webcam = False
    if self.combo.currentText() == 'yolov5':
        th = threading.Thread(target=self.detect_vid)
    elif self.combo.currentText() == 'centernet':
        th = threading.Thread(target=self.open_cam_centernet)
    elif self.combo.currentText() == 'faster-rcnn':
        th = threading.Thread(target=self.open_cam_faster_rcnn)
    elif self.combo.currentText() == 'retinanet':
        th = threading.Thread(target=self.open_cam_retinanet)
    elif self.combo.currentText() == 'ssd':
        th = threading.Thread(target=self.open_cam_ssd)
    th.start()
```

PART THREE



系统演示



PART FOUR



总结与展望

总结

在实验结果方面，YOLOv5算法模型对于口罩类型的识别和合理佩戴的检测都取得了良好的效果，并与其他算法进行了对比。在置信度和检测性能上，YOLOv5综合比所有参考算法都更具有该场景下的优势。

在小目标的检测上还有一定的提升空间，仍然可以通过进一步优化网络结构，或者选取更好的数据集，更多、更平衡的数据样本等来提升模型的检测精度。训练数据集由于硬件资源的限制设置得比较小。

结合其他技术可进一步提高检测的效率和准确性

- 1、人脸识别：将口罩佩戴情况与人脸信息相结合，可以实现精准识别。
- 2、语音识别：通过语音播报来提醒佩戴口罩的人员，增加监管效果。
- 3、智能报警：对于未佩戴口罩或佩戴不规范的情况，发送指令进行智能报警，提高检测的效果和时效性。
可以根据具体应用场景，组合这些技术，进行定制化设计。
- 4、硬件平台优化
某些场景下，需要更加灵活和高效的硬件平台，为特定场所进行设计，内置系统硬件，将监测程序集成到硬件中



THANKS

FOR YOUR WATCHING

