

# 边缘滤波计算下的图书馆座位识别系统研究

侯鹏<sup>1</sup> 李凯<sup>1</sup> 高娟<sup>1</sup> 吉彦平<sup>2</sup>

(1 陆军工程大学军械士官学校 湖北武汉 430000)

(2 河西学院 甘肃张掖 734000)

**【摘要】** 关于图书馆座位有效利用情况成为了管理者和读者关注的重要数据，目前图书馆座位信息主要基于读者刷卡、手机扫描二维码读取等方式获取，只能实现门禁出入管理，对于座位的有效利用基本依靠人工。面对激增的读者人群，高峰期刷卡进出图书馆耗时长，座位信息也无法精准掌握。本文通过基于图像识别的座位信息管理系统，利用边缘计算算法，有效的滤除无用信息，提高目标检测的实时性，大大缩短了高峰期进出图书馆的时间，确保读者信息安全可靠，能够精准掌握读者座位信息，实现较好的资源管理。

**【关键词】** 图像识别；座位管理系统；边缘计算

DOI: 10.18686/jyfzyj.v3i7.47351

## 图书馆选座系统需求探析

习近平总书记就读书重要性做了一系列的重要指示 [1-3]，提倡大家要多读书多学习，做到终生学习，终生读书，终生让自己面对生活的各种快乐。而城市或者学校中，图书馆是一个读书的最好场所，很多人希望来到图书馆能够有个舒适的位置，然后立即读书，这种情况是最理想的，但往往是因为没有座位或者座位被占，导致出现一些不方便的情况，在这种情况下图书馆订座系统就应运而生了，利用图书馆订座系统将图书馆座位的使用数据反馈到相应的平台上，读者可通过网络接收到实时数据，可直观了解图书馆的位置情况，并对座位进行预订。

## 边缘计算算法分析

国内现有的图书馆管理方法有：人工管理，校园卡等 [4-6]。目前各种图书馆管理系统有：基于云计算的图书馆管理系统，基于单片机的图书馆座位管理系统，基于二维码的图书馆座位管理系统等。由于本身的一些缺陷，这些系统都没有得到广泛的运用。人工管理模式效能低下，无法解决现有情况。图书馆是人员流动极大的公共场所，普通的云计算方式不能快速的处理边缘端设备产生的实时数据，面临资源紧缺以及服务延迟问题，这样数据的安全性得不到保障，本文提出一种基于边缘滤波平台的图书馆管理和目标检测和识别系统，这样可以有效提高大家对于图书馆的利用和安全性，实现图书馆有效应用。

## 1、目标检测相关技术

### 1.1 神经网络模型概述

目前采用深度学习的目标检测算法有一步算法 (one-stage) 和两步算法 (two-stage)。两步算法用 R-CNN 和 FASTR-CNN 等来表示。目标检测的第一步是通过提议找到候选区域，第二步是对候选区域中的目标进行分类，确定其类别。该算法的优点是精度高，识别准确，但缺点是识别速度慢，不能满足实时性要求。一步法直接识别输入图像，不生成候选区域。单次检测就可以得到目标类别的概率值和位置坐标值，直接得到最终结果。图书馆预约系统是为了实现实时的目标检测，主要研究目标检测的实时性，因此选择一步算法作为研究对象。

## 2、系统介绍及效果分析

### 2.1 系统整体介绍

图书馆摄像头采集头像，然后进行有效处理，通过识别就可以进入图书馆，使用预定座位。图 3.1 是图书馆座位管理系统整体架构图。

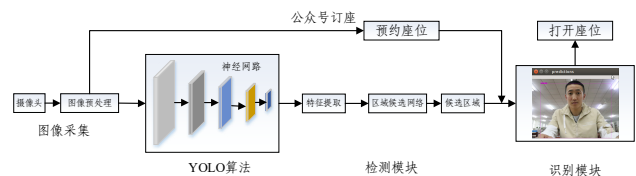


图 3.1 图书馆座位管理系统整体架构图

### 2.2 硬件介绍

本系统采用人工智能环境下的 Raspberry Pi 开发系统，结合各种控制模块，运用人脸识别采集信息时效性高，更加安全准确，有效保护学生信息安全，提高了图书馆的有效性。图 3.2(a) 是系统实物图。图 3.2(b) 是系统外观专利图。

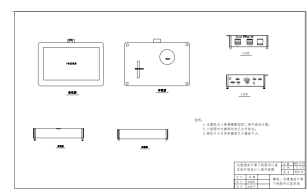
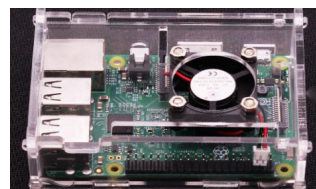


图 3.2 (a) 系统实物图

图 3.2 (b) 系统外观专利图

### 2.3 软件介绍

#### 2.3.1 摄像头显示

```
import cv2
import ipywidgets.widgets as widgets
import threading
import time
#设置摄像头显示组件
image_widget = widgets.Image(format='jpeg', width=500, height=400)
display(image_widget) #显示摄像头组件
```

第一步：设置摄像显示组件

第二步：视频格式转换

```
#bgr8转jpeg格式
import enum
import cv2
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

第三步：打开摄像头，图像参数设置

```

image = cv2.VideoCapture(0) #打开摄像头
# width=1280
# height=960
# cap.set(cv2.CAP_PROP_FRAME_WIDTH,width)#设置图像宽度
# cap.set(cv2.CAP_PROP_FRAME_HEIGHT,height)#设置图像高度
image.set(3,600)
image.set(4,500)
image.set(5,30) #设置帧率
image.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'))
image.set(cv2.CAP_PROP_BRIGHTNESS, 63) #设置亮度 -64 - 64 0.0
image.set(cv2.CAP_PROP_CONTRAST, 0) #设置对比度 -64 - 64 2.0
image.set(cv2.CAP_PROP_EXPOSURE, 2000) #设置曝光时间 1.0 - 5000 156.0

ret, frame = image.read() #读取摄像头数据
image_widget.value = bgr8_to_jpeg(frame)
    
```

2.3.2 OpenCV 人脸检测

第一步：导入必要的库

```

#导入必要的库
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets
from picamera.array import PiRGBArray
from picamera import PiCamera
from functools import partial
import multiprocessing as mp
import cv2
import os
import cv2 as cv
import threading
import time
    
```

第二步：显示摄像头组件

```

#设置摄像头显示组件
image_widget = widgets.Image(format='jpeg', width=500, height=400)
display(image_widget) #显示摄像头组件
    
```

第三步：转换图像格式

```

#图像格式转换
import enum
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
def f(x):
    return x
    
```

第四步：开始运行人脸检测程序



2.4 识别效果

静态图像识别效果如下，图 3.3 给出了目标检测结果，模型将目标框出，并给出了类别和置信度 (confidence)，可以看出，在图像距离较近，较为清晰的情况下，目标识别的效果比较好。

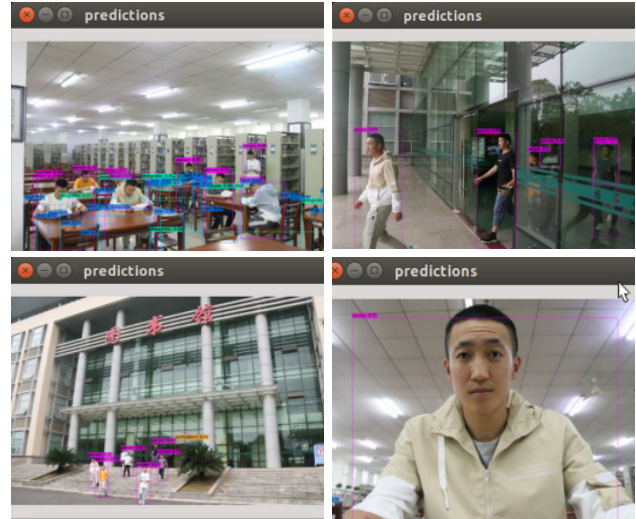


图 3.3 目标检测结果

3、总结

本文以树莓派硬件平台为依据，运用 YOLO 算法实现人脸识别与检查，运用在图书馆订座场景中，完成人员进出图书馆和订座的有效识别，实现了图书馆订座功能，该系统最图书馆管理有一定的实用性。

参考文献

[1] 解晓萌. 复杂背景下运动目标检测和识别关键技术研究:[博士学位论文]. 华南理工大学: 华南理工大学图书馆, 2012.

[2] 王守佳. 基于图像的人体检测跟踪和人脸识别的研究:[博士学位论文]. 吉林大学: 吉林大学图书馆, 2013.

[3] 马帅. 面向移动端的轻量级卷积神经网络分类算法研究:[硕士学位论文]. 华南理工大学: 华南理工大学图书馆, 2018.

[4] 蔡成飞. 基于人脸识别技术和边缘计算技术的智能系统研究:[硕士学位论文]. 浙江大学: 浙江大学图书馆, 2018.

[5] 昌伟. 基于深度学习的目标检测服务平台研究与实现:[硕士学位论文]. 北京邮电大学: 北京邮电大学图书馆, 2018.

[6] 陈睿敏. 手势识别关键技术及其在智能实验室中的应用研究:[硕士学位论文]. 中国科学院上海技术物理研究所: 中国科学院大学图书馆, 2017.