**课程设计说明书**

课程名称： 物联网数据处理综合实训

学 院： 计算机科学与工程学院

专 业： 物联网工程 班 级： 1,2

姓 名： 刘东临 学 号： 201601061413

姓 名： 亓乐 学 号： 201601061417

姓 名： 赵静 学 号： 201601061339

姓 名： 史邵冲 学 号： 201601061419

**山 东 科 技 大 学**

**2019 年 7 月 20 日**

目录

[第一章 概述 3](#_Toc13990247)

[1.1项目的介绍 3](#_Toc13990248)

[1.2项目的背景 3](#_Toc13990249)

[1.3问题重述 4](#_Toc13990250)

[1.4解决的思路 5](#_Toc13990251)

[1.5项目的目标 5](#_Toc13990252)

[第二章 需求分析 5](#_Toc13990253)

[2.1功能需求 5](#_Toc13990254)

[2.2性能需求 6](#_Toc13990255)

[2.3接口需求 6](#_Toc13990256)

[第三章 系统设计 7](#_Toc13990257)

[3.1 系统架构 7](#_Toc13990258)

[3.2 设计原则 8](#_Toc13990259)

[3.3 运行环境分析 8](#_Toc13990260)

[3.4 功能模块划分 10](#_Toc13990261)

[3.5 关键技术分析 10](#_Toc13990262)

[3.6 接口设计 11](#_Toc13990263)

[3.7 运行设计 11](#_Toc13990264)

[第四章 详细设计 11](#_Toc13990265)

[4.1 图像分割模块 11](#_Toc13990266)

[4.2 人物检测模块 11](#_Toc13990267)

[4.3 主模块 12](#_Toc13990268)

[第五章 系统实现与测试 12](#_Toc13990269)

[5.1运行界面截图 12](#_Toc13990270)

[5.2系统评测 14](#_Toc13990271)

[第六章 分析总结 14](#_Toc13990272)

[6.1系统总结 14](#_Toc13990273)

[6.2遇到的问题 15](#_Toc13990274)

[6.3解决方案 15](#_Toc13990275)

[6.4存在的问题 15](#_Toc13990276)

# 

# 第一章 概述

## 1.1项目的介绍

数字图像处理在国民生产中起到日益重要的作用，对其的研究也日益广泛和深入。OpenCV程序库作为数字图像研究的软件助手，具有简单易用、功能强大、移植方便等优越性能。本实验设计了一个基于OpenCV的运动目标检测跟踪实验平台，对从摄像头采集来的图像数据进行实时的分析，从而实现对运动物体的检测与跟踪。本实验旨在引导学生了解OpenCV的性能，学习OpenCV的使用方法；尝试并逐渐熟悉图像处理，特别是运动目标跟踪方面的知识；以及进一步提高Windows操作系统下使用python编程的能力。

总之，海水图像处理系统不仅实现了对OpenCV进行基本的初步理解，还实现了图像区域分割、人物检测、人群计数等功能，基本可以满足用户的相关需求。

## 1.2项目的背景

**1.opencv相关信息：**

opencv是hite严开源计算机视觉库。它由一系列c函数和少量c++类构成，实现了图像处理和计算机视觉方面的很多通用算法。

其重要特性包括:拥有包括300多个C函数的跨平台的中、高层API。它不依赖于其它的外部库——尽管也可以使用某些外部库。

openCv的优势在于:（1）纯c代码，源代码开放；（2）丰富的函数功能，强大的图像和矩阵运算能力；（3）平台无关性；（4）程序运行的实时性；（5）方便灵活的用户接口；（6）交互性及强大的扩展功能；（7）可嵌入性。

可见，作为一个基本的图像处理、计算机视觉和模式识别的开源项目OpenCv可以直接应用于很多领域，作为二次开发的理想工具。

**2.图像噪声的处理方面**

图像信号在产生、传输和记录过程中，经常会受到各种噪声的干扰，一般来说，现实中的图像都是带噪图像。通常在图像处理工作中，在边缘检测、图像分割、特征提取、模式识别等高层次处理之前，选用适当的方法尽量地去除噪声干扰是一个非常重要的预处理步骤。依据噪声产生的原因，将经常影响图像质量的噪声源分为三类:阻性元器件内部产生的高斯噪声;光电转换过程中的泊松噪声(椒盐噪声);感光过程中产生的颗粒噪声。噪声的均值表明了图像中噪声的总体强度。一般地，对噪声的描述采用统计意义上的均值与方差。

综合使用各种方法去除图像噪声，通过实验比较了这些方法所适用的场合，深入了解了其中的均值和中值滤波算法，兼顾质量和实时性的要求，将两者结合起来，采用一种改进的实时滑窗递归中值算法。

**3.运动目标检测方面**

重点学习了目前在实际应用中普遍采用的YOLO算法。

Yolo采用一个CNN网络来实现检测，是单管道策略，其训练与预测都是end-to-end，所以Yolo算法比较简洁且速度快。第二点由于Yolo是对整张图片做卷积，所以其在检测目标有更大的视野，它不容易对背景误判。其实我觉得全连接层也是对这个有贡献的，因为全连接起到了attention的作用。另外，Yolo的泛化能力强，在做迁移时，模型鲁棒性高。

最后不得不谈一下Yolo的缺点，首先Yolo各个单元格仅仅预测两个边界框，而且属于一个类别。对于小物体，Yolo的表现会不如人意。这方面的改进可以看SSD，其采用多尺度单元格。也可以看Faster R-CNN，其采用了anchor boxes。Yolo对于在物体的宽高比方面泛化率低，就是无法定位不寻常比例的物体。当然Yolo的定位不准确也是很大的问题。

Yolo与其它检测算法做了对比，包括DPM，R-CNN，Fast R-CNN以及Faster R-CNN。其对比结果如表1所示。与实时性检测方法DPM对比，可以看到Yolo算法可以在较高的mAP上达到较快的检测速度，其中Fast Yolo算法比快速DPM还快，而且mAP是远高于DPM。但是相比Faster R-CNN，Yolo的mAP稍低，但是速度更快。所以。Yolo算法算是在速度与准确度上做了折中。

## 1.3问题重述

本项目在Windows环境下基于Python语言，OpenCV的视频/图像处理的一个海水浴场人员与物体识别。

问题一：识别沙滩和海水，画出沙滩和海水分界线；

问题二：识别沙滩和海水上各有多少人；

在图像分割模块中，本系统需要解决如何识别什么是沙滩，什么是海水，以及如何调用图像分割函数。同时，本系统还需要解决海水沙滩分界线连续清晰的问题。

在人物检测模块中，本系统需要解决如何对人群和其他物品等进行区分。由于沙滩人数多、海水人暴露部分不唯一，因此必须设计合理的人群检测识别，使系统能够获得较精准的效果。

在主模块中，本系统需要解决各个模块之间的调用，以及人数统计等。

## 1.4解决的思路

我们项目在设计时采用模块化编程，将系统划分为3个主要模块，分别为图像分割模块、人物检测模块以及主模块。

本系统的整体设计思路主要可以分为如下几步：熟练掌握OpenCV的使用；完成图像处理的相应处理操作；实现海水沙滩分界线的绘制；完成人群的检测，最后分别统计沙滩和海水的人群计数。

## 1.5项目的目标

通过对图像分割模块、人物检测模块以及主模块的模块化编程，并将他们聚合为一个具备上述所有功能的强大的视频图像处理系统，实现一个灵活性好、覆盖面高、功能强大的图像处理与检测系统。

# 第二章 需求分析

## 2.1功能需求

功能性需求即软件必须完成哪些事，必须实现哪些功能，以及为了向其用户提供有用的功能所需执行的动作。软件需要能识别什么是海水什么是沙滩，并将海水和沙滩的分界线绘制在图像上，还要满足用户对人数检测的操作，在显示该图像时能够看到海水沙滩分界线的轮廓，并在画面中对人群进行检测标注。与此同时，本系统还支持对人数的检测，这主要包含两个方面：海水中的人与沙滩上的人。该项目主要包括3个主要功能：图像分割功能、人物检测功能以及人群计数功能等。本系统所支持的功能基本可以满足问题重述中的需求。

本系统可以分为3大部分，第一个子模块所包含的功能为图像预处理和颜色分割算法。图像增强(Image enhancement)是数字图像处理技术中最基本的内容之一，也是预处理方法之一。预处理过程一般包括：数字化、几何变换、归一化、平滑、复原、增强等。图像增强分为基于空间域的图像增强和基于频率域的图像增强。第二个子模块所包含的功能为人物识别和人物统计，实现沙滩和海水上人物的区分识别和人数计数等子功能。第三个子模块所包含的功能为连接各个功能模块，实现各个模块之间的调用。

本系统通过分析用问题的需求设计了这款针对沙滩和海水的图像处理软件，满足了问题的基本需求：识别沙滩和海水，画出沙滩和海水分界线；识别沙滩和海水上各有多少人。这在一定程度上满足了项目实现的目标，提供给用户良好的体验，使用户能够较为方便地进行相应的操作。

## 2.2性能需求

通过相关的分析，本系统应当是一款实时性较高的软件。其必须满足用户能够实时检测海水和沙滩上人数的需求。下面将根据用户的需求从如下几个方面阐述具体的性能需求。

1. 启动界面的性能需求：本系统在启动时，封面应当为高清晰度的图像，选用的色彩不宜超过256色，大小多为主流显示器分辨率的1/6大。整个系统的启动时间不应当超过4秒。
2. 操作界面的性能需求：软件框架设计应该简洁明快，尽量少用无谓的装饰，应该考虑节省屏幕空间，各种分辨率的大小，缩放时的状态和原则。
3. 图像的清晰性能需求;在显示沙滩海水分界线时应当保证分界线的清晰性，确保能够清晰地绘制到采集的图像数据。在使用时能够获得良好的体验效果。

## 2.3接口需求

本系统在设计时采用的是模块化的设计方法，因此，在对各个模块进行设计时必须保证各个模块对外接口的健壮性与可移植性。各个模块对外提供的接口既要实现能够提供基本的操作的功能，又要保证外界调用的便捷性。本系统所设计的3个模块主要为图像分割模块、人物检测模块以及主模块。本系统的各个模块之间的接口调用关系如图1所示：

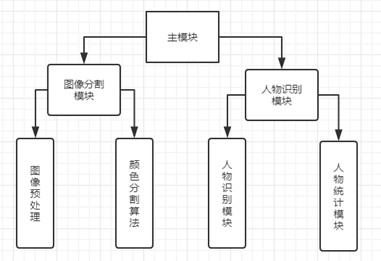


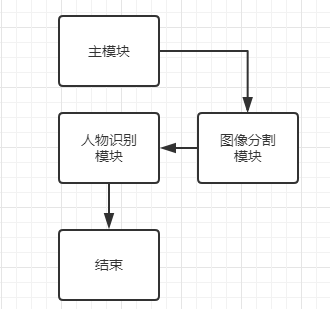
图1 模块接口调用关系图

1. 主模块接口
2. 图像分割模块接口
3. 人物识别接口

# 第三章 系统设计

3.1 系统架构

本系统采用模块化的设计思想，将本系统划分为3大模块：主模块、图像分割模块、人物识别模块。通过协调各个模块来实现沙滩图像的分割和人物检测的处理。本系统的整体框架是类似于金字塔的模型，主模块处于顶层，分别调用图像分割模块和任务识别模块来进行识别和分割处理。其具体的系统架构如下图所示：



系统架构图

3.2 设计原则

本系统在设计时考虑到日后的可移植性以及健壮性，故在设计时采用如下的设计原则：

1. 变量命名规则：本系统在设计时采用Linux系统中常用的命名方法：下划线命名方法。每个单词之间用下划线来进行分割。
2. 接口使用规则：本系统在设计时采用统一的接口使用方式。对于其调用的参数，包括操作的对象、操作的长度以及其他参数。一般情况下，操作的参数列表个数不超过5个。
3. 图片处理规则：本系统在对图片进行处理时考虑到原始图片在进行后期处理时需要重复使用，因此，所有对图片的处理对象均为图片的副本。
4. 模块独立规则：本系统在设计各个模块时充分考虑到模块之间的关联性。因此，各个模块对外呈现出一种低拟合的状态。各个模块均可以独立调用、调试、运行，彼此之间的影响很小。

3.3 运行环境分析

(1)整体分析

本系统为了能够保证系统拥有良好的兼容性，在设计时主要采用如下的运行环境以及开发工具：

运行平台：windows10操作系统

开发语言：python 3.6

软件开发工具：PyCharm

视觉库工具：OpenCV 3

(2)环境配置

本系统是利用pycharm开发工具进行开发，使用anaconda配置python的运行环境，下面将具体介绍安装的步骤。首先安装anaconda，从官网下载相应的安装包，下载网址为<https://www.anaconda.com/download/>。

选择合适的安装版本，这里选择windows版本。Python的版本为3.6。

python 安装opencv模块：

参考<https://blog.csdn.net/u011321546/article/details/79499598>

需要安装opencv和numpy：pip install opencv-python

pip install numpy

pip install -U scikit-image

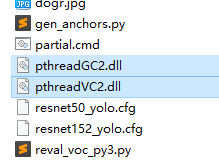
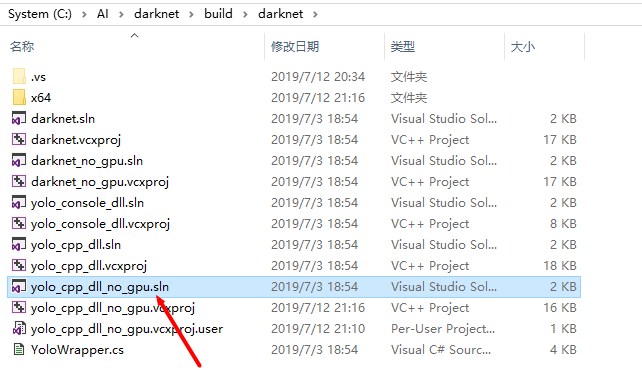
利用github上最热的物体检测模块darknet来进行人物识别，需要安装配置darknet：<https://github.com/AlexeyAB/darknet>

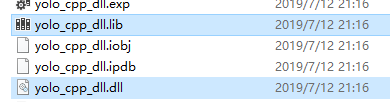
安装配置VS已生成相应的darknet的本机运行动态链接库dll。参考<https://blog.csdn.net/baidu_36669549/article/details/79798587>

下载项目

下载权值数据<https://pjreddie.com/media/files/yolov3.weights>

用VS打开，生成解决方案，得到dll

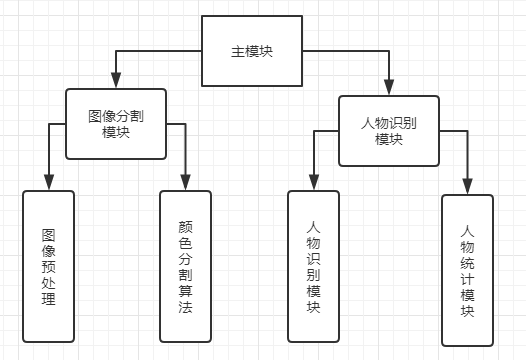




将生成的四个文件放到相应的文件夹下，之后可以调用darknet作为人物识别模块。

3.4 功能模块划分

本系统根据实际的使用情况，可以划分为2大主模块：图像分割模块、人物识别模块，每个主模块又可以划分为2个子模块：图像预处理模块，颜色分割模块； 人物识别模块，人物统计模块。本系统具体的功能模块划分图如下所示：



功能模块划分

本系统根据上述功能模块设计，各个模块之间既存在一定的联系有存在相互独立，这保证了系统的健壮性。每个功能模块之家通过相应的函数接口进行相应的调用，每个模块基本都需要启动一个新的线程来实现相应的功能，这可能需要线程的调度与通信。

3.5 关键技术分析

1. OpenCV开发技术：OpenCV一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。
2. 图像分割技术：常见的图像分割算法包括：阈值分割，边缘分割，颜色分割，分水岭算法，聚类分割等等。本次项目因为沙滩颜色和海水颜色等相差比较大，可以采用简单且有效的颜色分割和边缘分割。
3. 物体检测技术：常用的物体检测算法SIFT/SURF、haar特征、广义hough变换。本次项目通过直接调用darknet完成物体检测。darknet采用的物体检测算法是YOLO3。

3.6 接口设计

本系统在设计时采用分层模块化设计，各个模块之间通过API进行相应的调用，从而实现系统的各个功能。本系统在设计接口时采用低拟合的设计思想。各个模块之间存在的接口主要包括图像分割接口、人物检测接口以及主模块调用接口。这三个接口基本实现了系统的主要功能，下面见对这三个接口做具体的介绍。

1. 本系统在设计图像标注接口时考虑到图片在后期需要进行二次处理，因此，所有的操作均为图片的副本。全景标注模块所用到的图片均来自视频采集模块提供的接口。该接口主要包含两个参数：原始的图片、当前采集的图片编号。本系统只需要调用该接口即可实现图片的获取。
2. 本系统在设计物体检测接口时只设计了检测人，针对本项目因为只需要检测人数。并且可信度可控，在可信度较低时，可以检测比较多的人，因为情境中干扰项不多（大部分图片只有沙滩、海、人、太阳伞）如果用户想要获得较高的灵敏度，可以提高可信度。

本系统所设计的接口较为精简，方便其他模块之间相互调用。在一定程度上方便日后进行拓展。

3.7 运行设计

命令行运行。

# 第四章 详细设计

4.1 图像分割模块

图像分割模块是本项目第一个待解决的问题，本项目通过设置阈值找到对应的沙滩颜色，之后设置蒙版，与远算截取出沙滩颜色的图片。通过腐蚀膨胀消除细小边界，减少噪声。将图像转成灰度图，做二值化处理，继续通过开运算等去除噪声，使findContours获得更好的效果，更小的干扰。然后调用opencv的findFontours函数来找到轮廓，此时该边界轮廓就是沙滩海水的分割线。

**'''**

**ColorSegement：基于颜色分割算法，分割海水和沙滩**

**parameter：image=输入源图片，lower=颜色阈值下限，upper=颜色阈值上限**

**return：返回海水沙滩分割后的图片**

**'''**

**def ColorSegement(image, lower, upper):**

**# 读取图片文件**

**image = cv2.imread(image)**

**cv2.imshow("image",image)**

**# 根据阈值找到对应颜色，设置蒙版，与运算截取出沙滩颜色的图片**

**mask = cv2.inRange(image, lower, upper)**

**output = cv2.bitwise\_and(image, image, mask=mask)**

**# cv2.imshow("cut", output)**

**#去噪**

**#先腐蚀后膨胀，可以消除一些细小的边界，消除噪声**

**kernel=np.ones((2,2),np.uint8)**

**fushi=cv2.morphologyEx(output,cv2.MORPH\_OPEN,kernel,iterations=3)**

**# cv2.imshow(u'fushi',fushi)**

**pengzhang=cv2.dilate(fushi,kernel,iterations=3)**

**# cv2.imshow(u"pengzhang",pengzhang)**

**# 转成灰度图**

**gray = cv2.cvtColor(pengzhang,cv2.COLOR\_BGR2GRAY)**

**# 二值化，以方便边缘分割**

**ret, binary = cv2.threshold(gray,0,255,cv2.THRESH\_BINARY)**

**# 设置kernel，用开运算继续去噪**

**kernel = cv2.getStructuringElement(cv2.MORPH\_RECT,(10, 10))**

**opened = cv2.morphologyEx(binary, cv2.MORPH\_OPEN, kernel)**

**# cv2.imshow("open",opened)**

**# 膨胀，使其圆润**

**pengzhang2=cv2.dilate(opened,kernel,iterations=3)**

**# cv2.imshow("pengzhang2",pengzhang2)**

**# findcontour找轮廓**

**aa,contours, hierarchy = cv2.findContours(pengzhang2, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)**

**# 填充空洞，再找一次，可以去除空洞，使边界完整**

**tianchong = fill\_contours(pengzhang2, contours, h = 20, w = 50)**

**aa,contours, hierarchy = cv2.findContours(tianchong, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)**

**# print(np.size(contours)) # 得到该图中总的轮廓数量**

**# print(contours[0]) # 打印出第一个轮廓的所有点的坐标， 更改此处的0，为0--（总轮廓数-1），可打印出相应轮廓所有点的坐标**

**# 保证所有数据能够显示，而不是用省略号表示，np.inf表示一个足够大的数**

**# np.set\_printoptions(threshold=np.inf)**

**# # 若想不以科学计数显示:**

**# np.set\_printoptions(suppress=True)**

**# f = open("contours.txt","w+")**

**# f.write(str(contours[0]))**

**# f.close()**

**# print(hierarchy) # 打印出相应轮廓之间的关系**

**# 画出分界线**

**cv2.drawContours(image, contours, -1, (0, 0, 255), 2)**

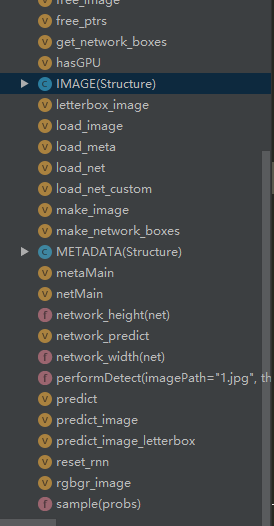
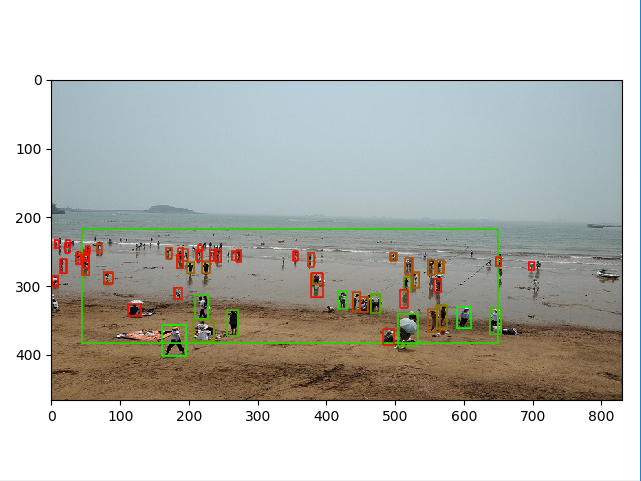
**# cv2.imshow("counter",image)**

**cv2.waitKey(0)**

**return image, contours[0]**

4.2 人物检测模块

人物检测是本实验的第二个待解决的问题，本项目通过调用darknet（YOLO3算法训练）。安装配置好darknet环境后（NoGPU版本），直接可以调用，需要修改一些函数，来增加一些功能，比如我们添加一个返回值（person数组），以统计人的坐标是在海里还是沙滩上。

4.3 主模块

主模块分别调用上述两个，先调用图像分割模块，画出分割线。后调用人物检测模块，检测出人，并且圈出人。在主模块通过比较，进行统计在海中的人数和在沙滩的人数各有多少并输出。

**from yolov3\_people.ColorSegment import \*  
from yolov3\_people.darknet import \*  
if \_\_name\_\_ == '\_\_main\_\_':  
 # 输入图片的路径  
 image = './image/4.jpg'  
 # 创建NumPy数组，设置沙滩颜色区间  
 lower = np.array([0,70,70]) # 颜色下限  
 upper = np.array([120,255,255]) # 颜色上限  
 img, contours = ColorSegement(image,lower=lower,upper=upper)  
 cv2.imshow("contour",img)  
 cv2.imwrite("contour.jpg",img)  
 # 调用performDetect函数，返回人物坐标数组  
 a,person = performDetect("contour.jpg")  
 # print(person)  
 PeopleInSea = 0  
 PeopleOnBeach = 0  
 for p in person:  
 xFootCoord = p[2][0]  
 yFootCoord = int((p[2][1] + p[3][1])/2)  
 FootCoordPoint = (xFootCoord,yFootCoord)  
 res = cv2.pointPolygonTest(contours, FootCoordPoint, measureDist=False)  
 if res == -1:  
 PeopleInSea += 1  
 else:  
 PeopleOnBeach +=1  
 print("Num of People In Sea: ",PeopleInSea)  
 print("Num of People On Beach: ",PeopleOnBeach)  
 cv2.waitKey(0)**

# 第五章 系统实现与测试

## 5.1运行界面截图

本系统根据需求实现了两个主要功能：找出海水沙滩边界，检测图片中中的人并统计人数。下面将具体展示本系统时实验效果，如下所示：



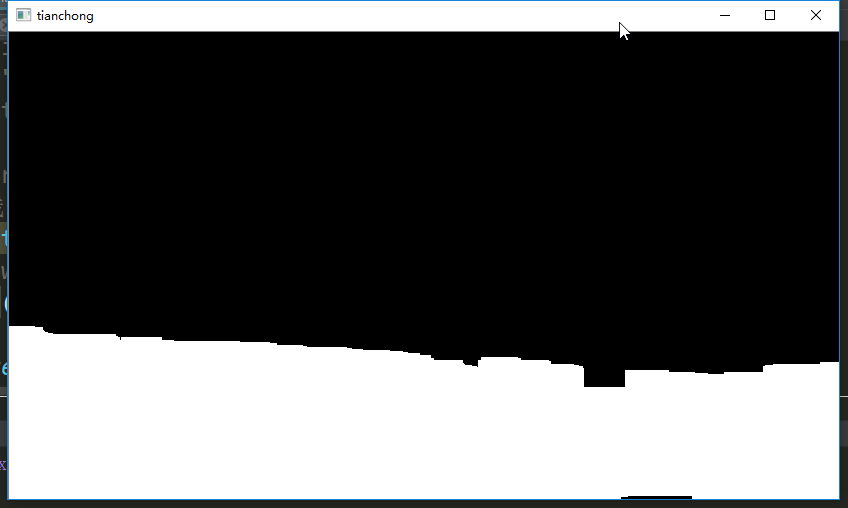
原图



通过颜色截取图片



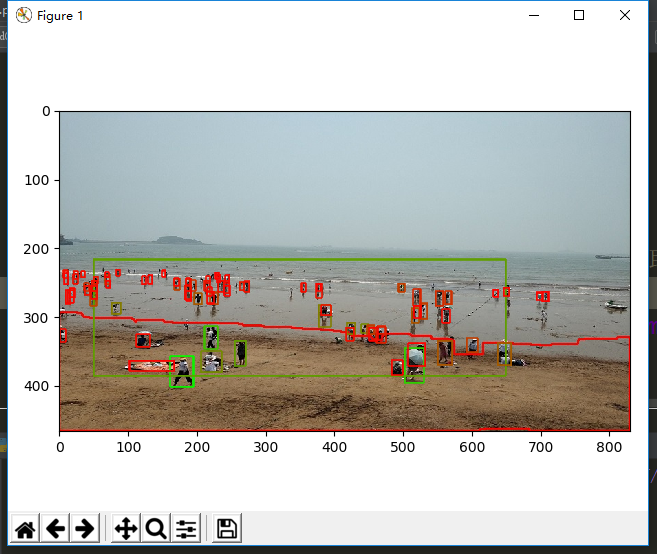
腐蚀膨胀处理



二值化与去噪处理



画出沙滩海水边界



人物检测

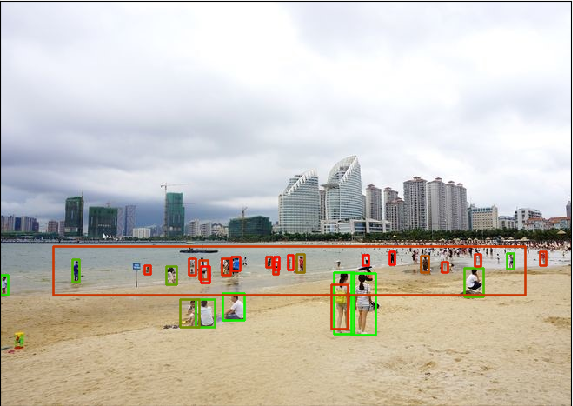
## 5.2系统评测

测试机处理器为Intel i5-5257U，内存为8G，运行环境为windows，机器位数为64位。具体的测试内容如下：

1. 测试沙滩分割效果。延时在0.5s左右。测试其他图片，在更改阈值后，有比较好的分割效果。



1. 测试人物识别效果。测试5张图片，对其中人物进行识别标记，效果较好。延时在3s左右（主要时间是画图的时间，识别时间较快）。



1. 普适性测试：用同样的代码测试不同的图片，发现效果并不是很好。测试不同图片时，需要改变阈值。
2. 误差分析：颜色分割要求阈值比较精确，且每张图片，其沙滩色的颜色阈值不尽相同，系统存在较大局限性。

# 第六章 分析总结

## 6.1系统总结

本系统是利用python语言进行开发的，利用opencv实现对图像的处理，设计了一款基于颜色分割和YOLO3的海水沙滩分割和海水沙滩人数检测系统。

## 6.2遇到的问题

本系统在实现的过程中遇到很多问题，最后通过上网查阅相关的资料解决了这些问题。其中，遇到的主要问题如下所示：

1. 图片用何种方式进行分割，效果比较好？
2. 分割前对图片需要进行何种预处理以达到更好的分割效果？
3. 如何通过python调用darknet项目实现人物识别和标记？
4. 人物识别和标记总是会忽略一些，怎么解决？

## 6.3解决方案

本系统针对上述问题分别提出了相应的解决方案：

1. 比较好的两种算法是分水岭算法和颜色分割算法，因为目标检测的是沙滩海水分界线，所以可以通过简单效果更好的颜色分割直接进行处理，先提取沙滩颜色的截图，进行去噪处理，之后进行二值化，找轮廓的方式直接可以找到沙滩海水分界线。
2. 通过腐蚀膨胀可以去除部分噪声，开运算去除部分空洞，二次膨胀处理可以使得图像更圆润，从而得到效果比较好的二值化图像，进行轮廓划线。
3. 查找资料后，发现可以通过VS2015打开darknet的相应模块，生成dll，进行调用。
4. 人物识别对于远景比较小的人物置信度较低，但是由于沙滩实际干扰物较少（事实上，大部分图片只有沙滩、海水、太阳伞和人），可以降低置信度，从而检测出更多的人。

## 6.4存在的问题

目前开发的系统实现了实训题目所要求的所有功能，处理效果较好。但同时也存在着一些问题，这些问题还有待解决：

1. 由于利用的是颜色分割，需要设置阈值，在不同图片上，可能效果不一定非常好，通常情况下需要更改阈值，以获得比较好的识别效果。
2. 对于图片中远景过小的人物，基本识别不出来。