



山东大学
SHANDONG UNIVERSITY

毕业论文(设计)

论文(设计)题目:

基于计算机视觉的快递领取辅助系统

姓 名_____张振民_____

学 号_____201920130242_____

学 院_____计算机科学与技术_____

专 业_____人工智能_____

年 级_____2019 级_____

指导教师_____赵海森_____

2023 年 5 月 25 日

山东大学本科毕业论文（设计）成绩评定表

学院：专业：年级：

学号		论文（设计）	分数	
姓名		总成绩	等级	
论文（设计）题目				
指导教师评语				
	评定成绩（分数）	×40%		
	教师签名：年 月 日			
答辩小组评语				
	答辩成绩（分数）	×60%		
	组长签名：年 月 日			

注：论文（设计）总成绩：分数=指导教师评定成绩（40%）+答辩成绩（60%）。
等级：优秀（90 分以上），良好（80-89 分），中等（70-79），及格（60-69），不及格（60 分以下）

摘 要

随着快递行业的发展，快递配送“最后一公里”问题并没有得到妥善解决，在学校、小区等快递量较大的快递驿站仍需要用户去领取快递。在快递驿站，快递被摆放在货架上，用户需要在指定的货架上找到自己的快递，用户平均需要浏览货架上将近一半的快递才能找到自己的快递，在一些摆放混乱的货架上，甚至需要用户翻找快递，这就导致用户在快递驿站寻找快递的效率低下。

为了解决快递驿站快递领取快递效率低下的问题以及提升用户体验，本文提出一种基于计算机视觉技术的快递领取辅助系统，本系统面向复杂的快递驿站场景，满足用户高效地寻找快递的应用需求，能够以直观可视的方式为用户描述快递位置，大大提高用户取快递的效率。

本文的主要研究内容包括两个方面，一是基于已有的计算机视觉技术，在快递摆放到货架上时，检测、识别并记录货架上的快递位置信息；二是采用安卓软件的形式，将快递位置信息呈现给用户，通过使用的可视化方法使用户可以更容易也更快得从货架上找到自己的快递，实现提高快递领取速度的目的，提升快递驿站的工作效率。

关键词：快递领取系统；计算机视觉；可视化；安卓软件开发

ABSTRACT

With the development of the express industry, the problem of "last kilometer" of express delivery has not been properly solved, and users still need to pick up express in schools, communities and other express stations with a large volume of express. In express stations, the express is placed on the shelf, and users need to find their express on the designated shelf. On average, users need to browse nearly half of the express on the shelf to find their express. In some disordered shelves, users even need to rummage through the express, which leads to the low efficiency of users looking for express in express stations.

In order to solve the problem of low efficiency of receiving express from Courier stations and improve user experience, this paper proposes an auxiliary system of receiving express based on computer vision technology. This system is oriented to the complex scene of Courier stations, meets the application needs of users to find express efficiently, and can describe the express location for users in an intuitive and visual way, which greatly improves the efficiency of receiving express from users.

The main research content of this paper includes two aspects. First, based on the existing computer vision technology, when the express is placed on the shelf, it detects, identifies and records the express location information on the shelf. The second is the use of Android software in the form of the express location information presented to the user, through the use of visual method users can more easily and faster to find their own express from the shelf, to achieve the purpose of improving the speed of express delivery, improve the efficiency of Courier station.

Keywords: Express Delivery System, Computer Vision, Optical Character Recognition, Android Software Development

目 录

第 1 章 绪 论.....	1
1.1 研究背景和研究意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	3
1.2 研究现状和发展趋势.....	3
1.2.1 研究现状.....	3
1.2.2 发展趋势.....	4
1.3 论文组织结构.....	4
第 2 章 快递领取辅助系统总体设计.....	5
2.1 快递位置信息获取综述.....	5
2.1.1 相关硬件介绍.....	6
2.1.2 相关软件介绍.....	6
2.2 可视化呈现综述.....	7
2.2.1 相关硬件介绍.....	7
2.2.2 相关软件介绍.....	8
第 3 章 快递位置信息获取.....	9
3.1 图像采集.....	9
3.2 取件码位置检测.....	10
3.2.1 根据特征匹配定位.....	10
3.2.2 改进后的取件码位置检测.....	11
3.3 取件码识别.....	16
3.3.1 图形矫正.....	16
3.3.2 光学字符识别.....	20
3.4 信息存储.....	21
3.5 实验结果评价.....	22
第 4 章 可视化呈现.....	23
4.1 软件功能实现.....	23

4.1.1 需求分析和设计.....	23
4.1.2 开发和编程.....	23
4.2 其他输出方式.....	27
4.3 本章小结.....	28
第 5 章 结束语.....	29
5.1 工作总结.....	29
5.2 工作展望.....	30
致 谢.....	31
参考文献.....	32

第1章 绪 论

1.1 研究背景和研究意义

当今社会，快递已经成为人们生活中不可缺少的一部分。然而，由于快递行业的飞速发展^[1]，快递包裹数量众多，快递驿站中快递领取效率低下的问题也愈发凸显。在此背景下，计算机视觉技术的应用成为了快递行业提高效率的重要途径^[2]。

1.1.1 研究背景

物流“最后一公里”配送是快递配送的最后一个流程，是指快递从快递驿站配送到用户手中。这一流程和快递用户息息相关，直接影响物流的效率和用户体验^[3]。

目前大多数快递驿站并没有实现“最后一公里”配送，仍采用快递配送到快递驿站（自提点），然后用户自提的方式解决。

快递有多种多样的类型。快递按照体积分类主要分为三类：其一是小件，如服饰、文件等，通常以快件袋包装；其二是中件，主要是鞋、包等稍大件的物品，同样用纸箱包装；其三是大件，常是被子、睡垫、学生行李等，重量超过 5kg，这类快件的包装不规则、体积较大，故快递常被就地堆放^[4]。

同样，快递在货架上的摆放形式也存在多种情况。大致可分为四类：

其一，快递全是大小差不多的矩形快递盒，排放有序，快递编号全部展现在外面，如图 1-1 所示。

其二，快递是大小不一的快递盒，摆放在一起，少部分快递编码被遮挡，如图 1-2 所示。

其三，快递是袋装的，是有规律的摆放，有遮挡和褶皱，如图 1-3 所示。

其四，快递随意摆放，且相互之间存在遮挡，快递的体积大小不一，快递类型也不一致，如图 1-4 所示。



图 1-1 快递盒大小相同，整齐摆放



图 1-2 快递盒大小不一



图 1-3 快递袋堆叠摆放



图 1-4 各种快递杂乱摆放

在这种复杂的快递领取场景下，用户在货架上找到自己的快递平均就需要浏览将近一半的快递。快递码不完全裸露在外面的情况下，甚至需要用户翻找快递。

这就导致在学校、小区等快递量较大的地方，用户在快递驿站寻找自己的快递效率低下。

1.1.2 研究意义

本文提出一种基于计算机视觉技术的快递领取辅助系统，该系统以取件码的位置代表快递的位置，使用相机对快递货架上的快递包裹进行拍摄，并进行取件码检测识别并记录取件码的位置信息，同时给用户提供语言描述和图像快递位置展示的可视化方式，解决用户在货架上查找快递慢的问题，使得用户能够直观快速的找到自己快递所在的位置，能够使快递领取更加便捷、高效。

1.2 研究现状和发展趋势

本节以本校菜鸟驿站的快递领取流程为例，介绍快递领取的研究现状和发展趋势。

1.2.1 研究现状

以本校菜鸟驿站的快递领取流程为例^[5]：菜鸟驿站营业时间为上午 10:00—下午 7:00，由快递员完成快递的分拣工作。快递按体积大概可分为三类，参考 1.1.1。当快递送到菜鸟驿站后，会根据快递类型和快递编号生成一个取件码，取件码由三段数字组成：第一段是货架号，用两位数字表示，第二段描述货架的第几层，用一位数字表示，第三段是快递编号后四位，用四位数字表示，取件码如图 1-5 所示。由快递员将取件码粘贴在快递包裹上。同时，学生会收到一条快递到达驿站的取件信息。在菜鸟驿站软件上同样会显示取件码等信息，提示用户取件。用户根据取件码到快递驿站的指定位置取出快递，然后到快递驿站的快递签收系统，通过快递单上的一维码和菜鸟软件提供的用户身份码确认取出快递，完成快递签收工作。



图 1-5 取件码格式

在整个快递领取过程中，用户按照指示找到对应的货架后需要在货架上找到自己的快递，这个过程需要用户平均浏览将近一半的快递，影响了快递领取速度。

1.2.2 发展趋势

目前，快递驿站提供的加速快递领取方法有使用快递柜存放快递^[6]和基于无人车进行“最后一公里”配送，直接配送到楼下。但是，二者也都存在一定的缺陷，比如快递柜占地面积大，快递驿站没有足够的空间设置足够的快递柜来存放所有待领取快递；无人车不能派送大件快递，无人车单次派送快递数量有限，不能为所有用户提供服务，无人车投入较大等。因此，使用货架存放快递依然是目前的主流方式。

因此，本文提出的基于计算机视觉技术的快递领取辅助系统，仍是一个重要的研究方向。

1.3 论文组织结构

本文分为 5 章，分别从研究背景和意义、研究现状和发展趋势、快递领取辅助系统概括和系统具体方法实现几个方面进行阐述。在本节中将概述每一章节的主要内容。

第一章为绪论，主要介绍论文的研究背景和研究意义、研究现状和发展趋势以及论文组织结构。

第二章介绍快递领取辅助系统的总体设计，概括地介绍本系统的两大内容：快递位置信息获取和可视化呈现，以及二者之间的关系。

第三章介绍快递位置获取的实现方法，包括取件码位置检测，取件码识别和取件码信息存储。

第四章介绍可视化呈现的实现方法，包括手机软件的实现和其他输出方式的实现。

第五章总结本文工作，提出工作展望。

第2章 快递领取辅助系统总体设计

基于计算机视觉的快递领取辅助系统的主要包含两个部分，一个是快递位置获取，以取件码的位置代表快递的位置，通过相机对快递货架上的快递进行拍摄，对图像进行处理，将取件码位置信息以特定的形式存储到数据库中。另一个是可视化呈现，提供文字描述和图像快递位置展示的可视化功能，给用户提供良好的领取快递体验。

快递位置获取将快递位置信息存储到数据库，可视化展示根据数据库中的快递位置信息，使用简洁高效的可视化方式呈现给用户。二者之间的关系如图 2-1 所示。

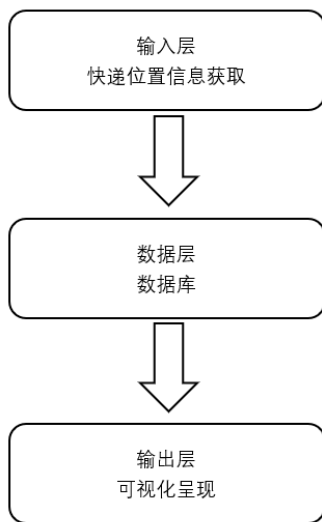


图 2-1 系统总体功能图

2.1 快递位置信息获取综述

快递位置信息获取基于已有的计算机视觉技术实现^[7]，包括特征匹配，阈值分割，光学字符识别等。快递位置信息获取是快递领取辅助系统的输入部分，可划分为图像采集、图像处理和信息获取、信息存储四个步骤。快递位置信息获取流程如图 2-2 所示。

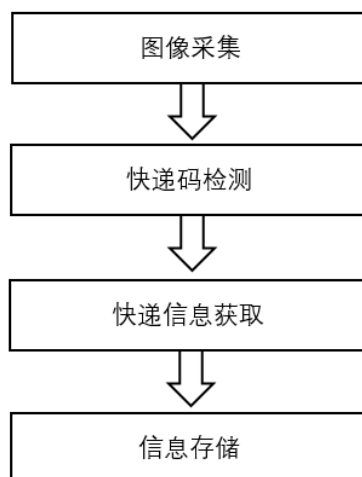


图 2-2 快递位置信息获取流程

快递信息获取主要用到的硬件设备有相机和计算机，相机用来采集图像，计算机对图像进行处理，得到快递位置新信息^[8]。

快递信息获取使用到的软件主要有 PyCharm 和 OpenCV，基于 python 语言实现。

2.1.1 相关硬件介绍

快递位置采集拍摄使用的设备为一个联想摄像头，光源是自然光，根据测试，均满足工作要求，相机参数：联想 WEC02 摄像头，传感器为 1/2.9 英寸，200 万像素，分辨率为 1080P/30fps。



图 2-3 联想摄像头

2.1.2 相关软件介绍

快递位置获取使用的软件和算法介绍如下：

1. PyCharm

PyCharm 是一款专门针对 Python 开发的集成开发环境（IDE），由 JetBrains 公司开发。PyCharm 具有许多效率工具和功能，包括代码自动补全、调试器、语法检查器、版本控制系统等，使得编写高效、可靠且易于维护的 Python 代码

变得更加容易。

2. OpenCV

OpenCV 是一个跨平台、开源的计算机视觉和机器学习库，其主要目的是提供通用的计算机视觉算法，并让开发者在进行视觉任务时不必过多关注底层实现细节。OpenCV 基于 C++ 编写，但同时也提供了一些 Python、Java 和 MATLAB 的接口，使得开发者可以在这些语言中调用 OpenCV 提供的函数进行编程。

2.2 可视化呈现综述

可视化呈现主要使用功能手机软件进行可视化。

本文利用手机软件的形式实现快递查询和快递位置展示功能。在高度信息化的今天，手机已经成为人们生活必不可少的一件设备^[9]。手机软件（MobileApp）是指在手机等移动设备上运行的应用程序，它相比传统的桌面应用程序拥有以下四个优势：便携性，移动化，社交化以及快捷支付。

总之，手机软件已成为移动互联网时代不容忽视的重要组成部分，它通过其便携性、移动化、良好的用户体验和社交化等特点，改变着人们工作和生活方式，并在多个行业产生了广泛的应用。

2.2.1 相关硬件介绍

安卓软件开发使用的手机参数如图 2-4 所示。



图 2-4 手机参数

2.2.2 相关软件介绍

软件开发使用的是 Android studio, Android Studio 是 Google 开发的一款安卓开发环境,也是现在最主要的一款 Android 应用程序开发工具。它提供了一系列的工具和模板来帮助开发者更加高效地创建 Android 应用程序。以下是 Android Studio 的主要特点:

1. 智能化代码编辑器: 支持实时代码分析,拥有自动完成功能,简洁清晰的 UI 可以帮助快速识别代码错误。

2. 集成的调试工具: Android Studio 提供多种模拟器来测试应用,还能够监控 APP 的性能,并可视化展示资源使用情况。

3. 支持多渠道发布: Android Studio 内置类似 Gradle 的构建工具,使得基于一个源码包可以编辑出适应多个平台的产品形态。

4. 插件支持: 可以通过下载第三方插件扩展 Android Studio 的功能,包括代码静态分析、UI 调整等功能。版本控制与集成: Android Studio 支持版本控制平台 Github 及其它 git 服务的集成,让团队协作开发变得容易。

总之,Android Studio 是一个非常强大的 Android 开发工具,它不仅提供了编写代码所需的一切工具和设施,而且具有友好的用户界面和一个强大的社区支持,使得开发者可以更专注地编写高质量的 Android 应用程序。软件开发界面如图 2-5 所示。

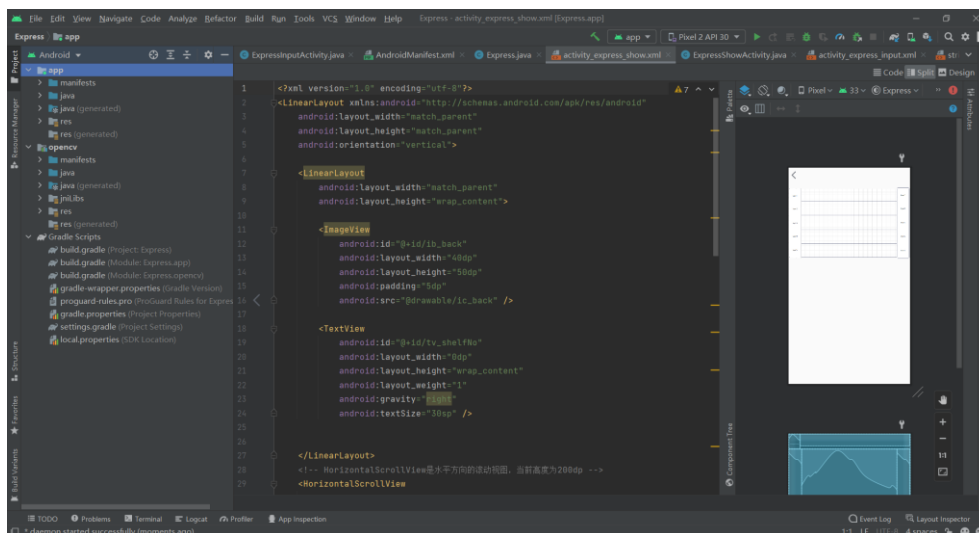


图 2-5 软件开发界面

第3章 快递位置信息获取

快递位置获取可以分为四个部分，图像采集、取件码检测、取件码识别、信息存储。

3.1 图像采集

根据货架号和货架层号将货架分为不同的处理单元，比如第 3 号货架，第 5 层，则第三号货架第五层采集到的图像作为一个处理单元，将识别到的快递信息存储到对应的文件夹中，文件夹以“货架号+层数”命名，并实时更新处理单元对应的采集到的图片。这种区域的划分可以在图像采集阶段实现。图像采集将相机摆放到正对货架的位置，货架样式如图 3-1 所示。以五层货架为例，每一层货架作为一个处理单元，所以在图像采集时需要针对不同货架区域进行划分，采用的方法是人工标注。

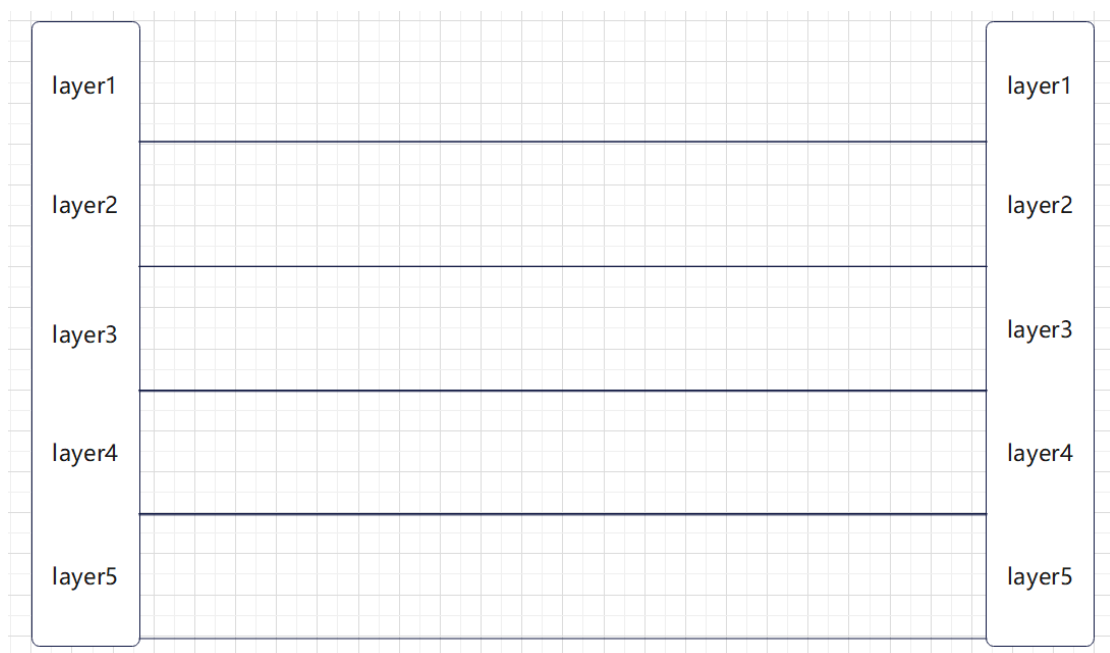


图 3-1 货架样式

管理员运行初始化程序，使用鼠标选中每一层的范围，然后将矩形框的左上顶点和右下顶点坐标存储到配置文件中，即完成一个货架的设置。初始化程序使用的主要方法：通过使用 OpenCV 提供的鼠标回调函数 `cv2.setMouseCallback (WindowName, onMouse)`，对窗口内的鼠标进行监听，第一个参数是窗口名称，是一个字符串类型，第二个参数是鼠标信号处理函数。`onMouse` 函数当鼠标左键

按下时记录左顶点坐标，鼠标左键松开手记录右下顶点坐标，管理员使用鼠标选中对应区域，即可得到每一层货架对应区域矩形信息。

3.2 取件码位置检测

取件码位置检测^[10]是通过对图像进行处理得到快递码所在区域。通过调研快递分拣方面的内容，了解到了两种快递码位置检测方法：根据特征匹配定位^[11]和根据二维码定位^[12]。

3.2.1 根据特征匹配定位

基于特征检测^[13]。以菜鸟驿站为例，每一层上的快递包裹上的取件码的第一段和第二段都是一样的，分别是货架号和层号。取件码如图 3-2 所示。该取件码是位于 33 号货架第 2 层的一个快递包裹。

使用特征匹配，首先需要制作特征匹配模板，即货架号和层号对应的图片，以第 33 号货架第 2 层为例，模板如图 3-3 所示。

然后分别对模板图像和待检测图像进行 sift 特征检测。OpenCV 提供了调用函数：`cv2.SIFT_create()`创建一个特征检测器，然后使用 `sift` 分别对模板图像和待检测图像继续特征检测。对两张图像上检测出的特征点，使用 `bf.knnMatch()`函数计算特征向量之间的距离，选取符合条件的对应特征点，特征匹配结果如图 3-4 所示。以本图为例，共检测得到 12 对匹配的特征，根据这 12 对特征的位置，可以计算出映射变换矩阵，即可将目标图像中的模板区域标出。然后根据模板区域和第三段取件码区域是相邻关系，即可标出第三段取件码区域。



图 3-2 菜鸟驿站取件码格式

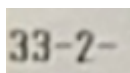


图 3-3 特征匹配模板

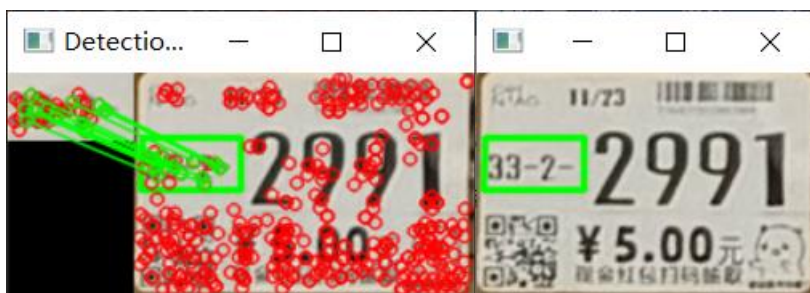


图 3-4 特征匹配结果

3.2.2 改进后的取件码位置检测

根据特征匹配定位、根据二维码定位这两种方法,对于相机精度要求非常高。因为相机需要拍摄整个货架上的内容,根据相机视野计算公式:

$$\text{相机视野} = \frac{\text{物距} \times \text{相机传感尺寸}}{\text{焦距}} \quad (3-1)$$

转换得到物距公式为:

$$\text{物距} = \frac{\text{相机视野} \times \text{焦距}}{\text{相机传感器尺寸}} \quad (3-2)$$

在相机确定的情况下,想要获得更大的相机视野就需要更大的物距,实际测量得到的物距是 1.2 米左右。在这种情况下就无法清晰的识别得到二维码、货架号层号。实际采集图像如图 3-5 所示。



图 3-5 摄像机拍摄的货架上的快递

实际裁剪得到的取件码区域如图 3-6 所示。由于整个取件码图像中包含的像素过少,使用特征匹配无法检测到其中的特征点,也就无法进行特征匹配。同样,由于二维码模糊以至于无法识别,也无法实现定位功能。

以图 3-5 所示货架快递为例,介绍最终采用的取件码位置检测流程。首先将图像转化为灰度图像,如图 3-7 所示。先对图片进行一个高斯滤波,使用函数 c

`cv2.GaussianBlur(image, (3, 3), 0)`，然后将图片转化为灰度图像，使用函数 `cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)`。



图 3-6 实际裁剪得到的取件码

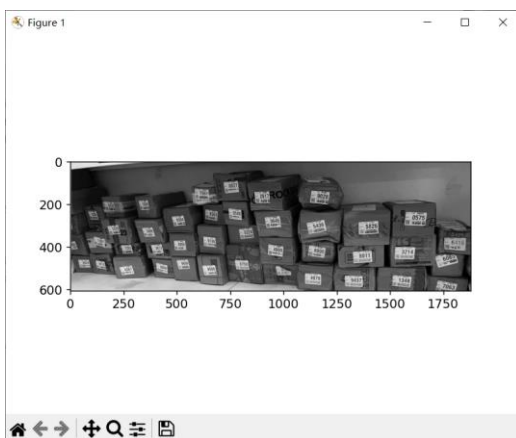


图 3-7 货架包裹灰度图像

然后对图像制作一个灰度直方图，如所图 3-8 示，使用函数 `gray_image.ravel()` 得到图像的像素分布，然后绘制灰度直方图，使用函数 `plt.hist(image_ravel, 256, [0, 256])`。对该灰度直方图进行分析发现，图像中有两个峰值，适合使用双阈值 OTSU 二值化进行阈值分割。

双阈值 OTSU 阈值化^[14]是由日本学者大津提出的，是图像分割领域的经典算法。经典 OTSU 算法思想是根据直方图确定阈值 K 将灰度图像分割成两部分，即目标 O 和背景 B ，使得 O 与 B 的类间方差最大。

其目标 O 与背景 B 之间的类间方差定义：

$$e^2(K) = P_a(\mu - \mu'_a)^2 + P_b(\mu - \mu'_b)^2 \quad (3-2)$$

其中 μ 是图像的灰度均值， μ'_a 和 μ'_b 分别是目标 A 和背景 B 的均值。当 K 的取值使得 $e^2(K)$ 最大时， K 即是最佳阈值。

通过使用函数 `cv2.threshold(gray_image, 125, 255, cv2.THRESH_BINARY)`

即可使用双阈值 OTSU 阈值化计算最优的图像分割阈值，改图计算结果为 125，并且会根据该函数对图像进行二值化处理，即可得到二值化后的图像，如图 3-9 所示。

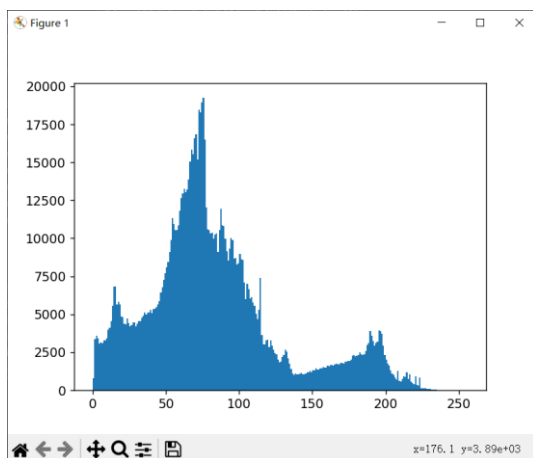


图 3-8 灰度直方图

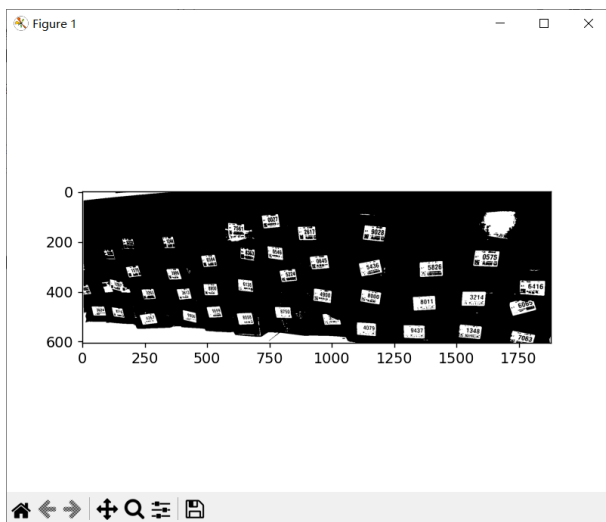


图 3-9 阈值化后的二值图像

然后需要对图像进行一系列的腐蚀和膨胀，消除噪声并使取件码白色区域联通。

腐蚀和膨胀^[15]都是基本的形态学操作，腐蚀和膨胀是针对白色部分而言的。对应到具体的图像中，如图 3-9 所示，白色部分就是指取件码部分。

在计算机视觉领域，腐蚀和膨胀是一些常用的形态学操作，常用于图像处理和图像分析中。腐蚀操作是指在图像中连续的像素点被取最小值，从而使图像中的边缘或细节变得更加明显。腐蚀操作通常会用于去除图像中的噪声或平滑图像中的细节特征。膨胀操作则是与腐蚀相反的操作，它是将图像中各点的像素值用最大

值取代，从而使图像更加明显。膨胀操作通常被用于图像中的腐蚀、使图像中的特征更为突出。在实际应用中，腐蚀和膨胀操作常常被组合使用起来，以实现更具体的应用场景，例如图像分割、图像纹理分析等。

在计算机视觉领域，开操作和闭操作是形态学操作的两种基本形式。它们都是结构元素和图像进行的像素级操作，可以改变图像的形状和大小，用于图像处理，分割和分析中。

开操作是指先进行腐蚀操作，再进行膨胀操作，这对去除小块和孔洞很有效。它可以消除图像中的噪声和细节，同时保留图像的边缘特征。在实际应用中，开操作可用于图像去噪和背景分割。

闭操作是指先进行膨胀操作，再进行腐蚀操作。它可以填充图像中的空洞和连接间断的区域，可用于去除小缺陷和连通图像的分离块。闭操作会使较小的区域消失，同时保留图像的整体结构。在实际应用中，闭操作常用于图像增强和形态学重建。

总之，开操作和闭操作在计算机视觉领域经常被应用，可以用于对图像进行预处理和后处理，提取图像的特征和信息，有效地提高图像处理的准确率和效率。

本例中，先使用 20×6 的矩阵对 x 方向进行闭操作（抑制暗细节），然后用 1×20 的矩阵对 y 方向进行开操作，使图像在 y 方向连通，最后使用一个 9×9 的滤波核图像进行中值滤波，主要是为了去除噪声。处理结果如图 3-10 所示。

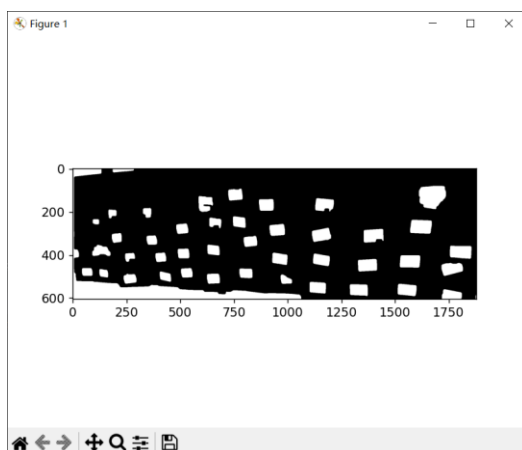


图 3-10 腐蚀膨胀后的二值图像

根据腐蚀膨胀的结果再对取件码区域进行边缘检测。边缘检测^[16]是计算机视觉领域的一种常用技术，旨在从图像中准确的识别出物体的轮廓线。它是图像分

析中最关键的部分之一，因为边缘检测可以得到物体的轮廓。常用的边缘检测算法有 Sobel 算子^[17]（Sobel operator）、Canny 边缘检测、拉普拉斯算子（Laplace operator）等。其中 Canny 边缘检测是较为常用且较为成熟的算法。以 Canny 边缘检测为例。Canny 边缘检测算法的操作步骤通常包括以下几个步骤：

1. 高斯滤波：用于平滑图像，以压制噪声。
2. 计算图像梯度：使用 Sobel、Prewitt 等算子计算每个像素点的梯度值和方向。
3. 非极大值抑制：在梯度方向上进行比较，但只有在局部的极大值处才能检测到边缘。
4. 阈值处理：将检测到的边缘像素点根据给定的高低阈值进行筛选和连接，形成完整的边界。

Canny 边缘检测算法具有检测精度高、定位准确、抗噪声能力较好等优点，在图像处理领域被广泛应用。

OpenCV 中提供了直接进行边缘检测的函数，`cv2.findContours(image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)`，`cv2.RETR_EXTERNAL` 表示只检测外轮廓 `cv2.CHAIN_APPROX_SIMPLE` 压缩水平方向、垂直方向、对角线方向的元素，只保留该方向的终点坐标。例如一个矩形轮廓只需要四个点来保存轮廓信息。`contour` 记录轮廓信息，`hierarchy` 记录轮廓层级关系信息。边缘检测结果如所示。

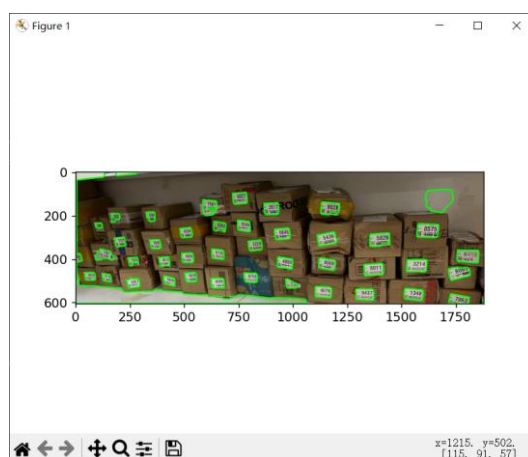


图 3-11 边缘检测结果

3.3 取件码识别

取件码识别完成的主要任务是根据上一步取件码位置检测得到的取件码区域识别图中的取件码，可以分为两个步骤实现，分别是图像矫正^[18]和光学字符识别。

3.3.1 图形矫正

首先，将取件码位置检测到的取件码轮廓使用最小外接矩形框出。使用矩形框出待选区域操作是 OpenCV 里最常用的操作，其中最为常见的就是包围框（Bounding Box）和旋转矩形（Rotated Box）。其中包围框是较为常见的，对应 OpenCV 中的 `cv2.boundingRect(item)`，使用正矩形框出物体，一般多用在目标检测中。使用包围框框出目标物体，这种操作比较简单，但是通常框中也会有一些其他的区域。其次就是使用旋转矩形，也叫最小外接矩形，对应 OpenCV 中的 `cv2.minAreaRect(contour)`，用来使用旋转矩形最大限度的框出目标物体，减小背景干扰，在 OCR 任务中较为常用。如图 3-12 所示，红色矩形框为包围框，绿色矩形框为最小外接矩形。

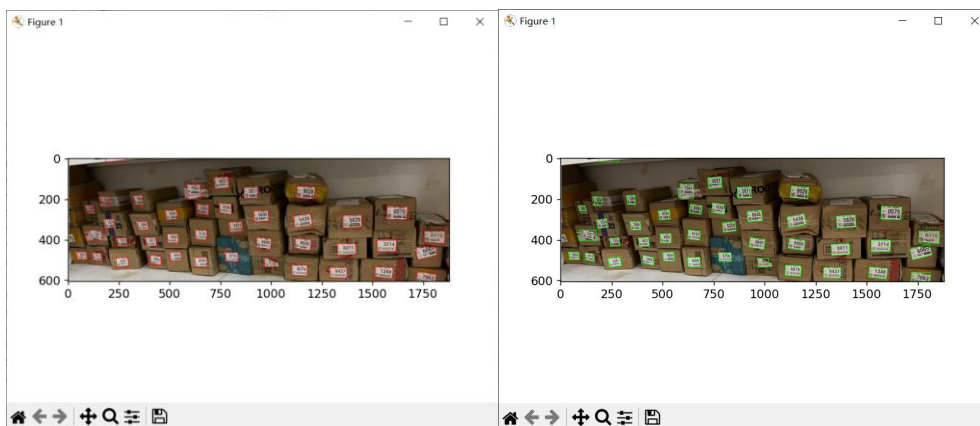


图 3-12 包围框（左）和最小外接矩形（右）

由于要进行取件码检测，所以在这里使用最小外接矩形框出取件码位置，减少其他因素干扰。`minAreaRect()`返回了所需区域的最小斜矩形的参数，与包围框直接返回四个顶点的坐标不同，最小外接矩形返回的是矩形的 $((x, y), (w, h), angle)$ ，对应了矩形的中心，宽度，高度和旋转角度。

旋转角度 `angle` 是水平轴（`x` 轴）逆时针旋转，与碰到的矩形的第一条边的

夹角。并且这个边的边长是 **width**，另一条边边长是 **height**。也就是说，在这里 **width** 与 **height** 不是按照长短来定义的。在 **OpenCV** 中，坐标系原点在左上角，相对于 **x** 轴，逆时针旋转角度为负，顺时针旋转角度为正，所以函数 **minAreaRect()** 返回的角度范围是 $[-90^\circ, 0^\circ]$ 。想象一个平放的长方形，调用 **minAreaRect()** 返回的角度为 -90° 。如果我们旋转图像，直到矩形树立起来，这是调用 **minAreaRect()** 得到的角度依然是 -90° 。返回参数值说明如图 3-13 所示。

使用最小外接矩形函数 **minAreaRect()** 将每一个轮廓包裹起来，结果如图 3-14 所示。然后要进行图像矫正。

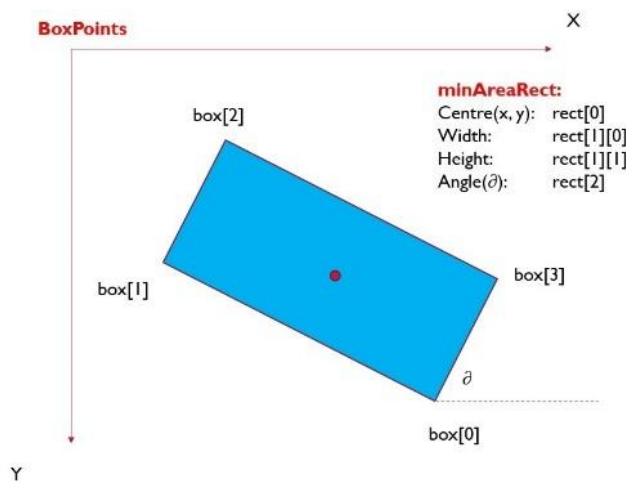


图 3-13 minAreaRect 函数返回值说明

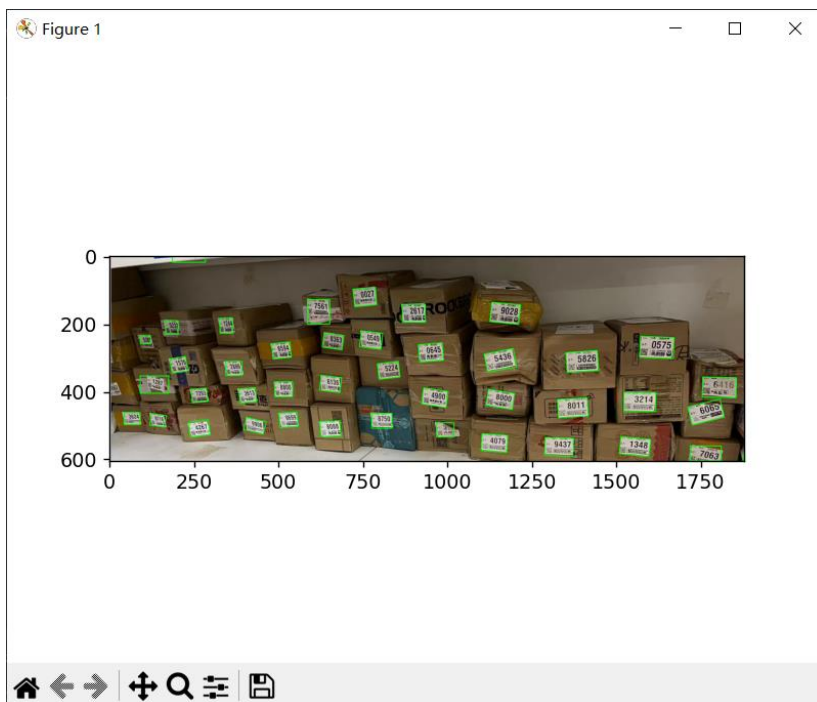


图 3-14 最小外接矩形

常用的矫正方法有仿射变换和透视变换。

仿射变换(Affine Transformation) 是一种二维坐标到二维坐标之间的线性变换，变换前后图形的平直性和平行性不会发生改变，如图 3-15 仿射变换所示。

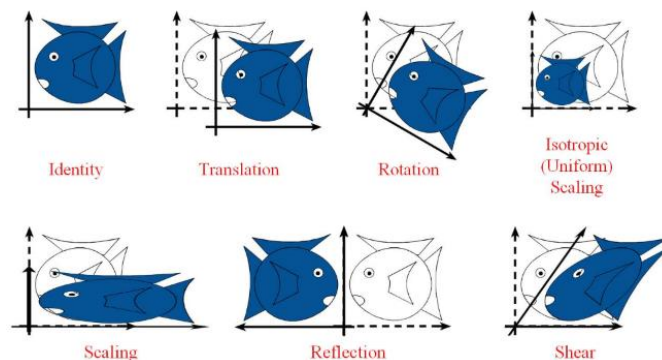


图 3-15 仿射变换

仿射变换计算过程：基于旋转矩形的中心和角度计算得到一个变换矩阵。根据取件码的特征，一个正确识别的取件码是宽度大于高度的。计划得到的取件码是一个正的矩形。因此需要判断一下 `angle`, `width` 和 `height`，保证旋转后的矩形是横着的矩形。以旋转矩形的中心为基准点，对整张图进行旋转，这里旋转的实现是基于仿射变换实现的。由于旋转之后矩形的中点坐标是不变的，以中心为基础，通过 `width` 和 `height` 抠出正矩形。如果不做边长和角度的判断，则只会沿是 `x` 轴的顺时针方向做相同大小角度的旋转，不能保证旋转后的视角是正确的视角，如图 3-16 所示。

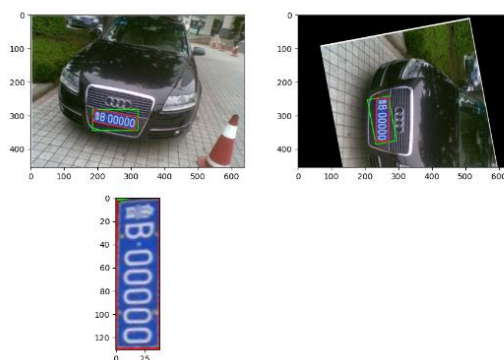


图 3-16 不合适的仿射变换

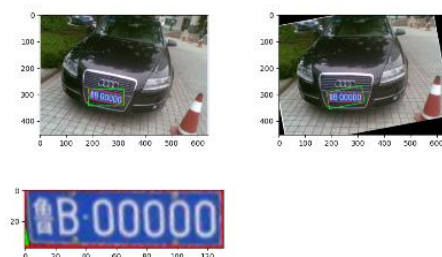


图 3-17 合适的仿射变换

根据任务目标的类型，做边长和角度的判断并进行相应的调整，可以保证旋转后的视角是正确的视角，如图 3-17 所示。

透视变换，也是本程序中采用的方法，透视变换是直接将旋转矩形的四个顶点映射到正矩形的四个顶点。透视变换也称为投影变换，是以投影的方式将图片投影到新图像平面，如图 3-18 所示。

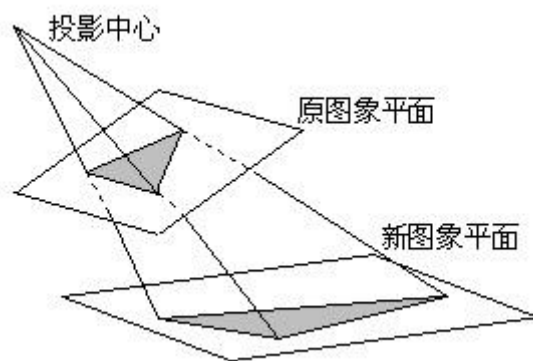


图 3-18 透视变换

基于矩形的四个顶点和想要抠出的正矩形的四个顶点得到一个变换矩阵。这里我想要得到的取件码是一个横着的正矩形，因此需要判断一下 width 和 height，将映射的平面定义为一个横着的正矩形。通过透视变换，将四个点组成的平面转换成另四个点组成的一个平面，以此抠出正矩形。透视变换结果如图 3-19 所示。



图 3-19 透视变换结果

3.3.2 光学字符识别

接下来就要对每一个取件码进行光学字符识别。

OCR^[19] 全称 Optical Character Recognition（光学字符识别），是一种用于将印刷体或手写文字数字化并储存到计算机中的技术。其主要工作就是将图像中的字符数据转化为可编辑或者可搜索的文本格式。OCR 的基本原理是通过对图像进行处理，把图像中的字符数据和排版结构还原成电子文件，并保存在计算机上。OCR 技术通常包含以下四步：

1. 图像预处理：对于输入的原始图像需要经过预处理来消除图像所含有的噪点、去除阴影、平行旋转等操作，以便得到更加规范的图像数据。
2. 字符分割：OCR 需要对所分析的图像中字符进行边界定位和划分，确保每个单独的字符都能被正确识别出来。
4. 特征提取：对于经过字符分割后得到的单独字符需要进行特征提取和编码，描述这些字符的形状和纹理信息，以便于后续识别。
5. 字符识别：根据图像特征提取出来的信息来对字符进行识别，并将识别结果转换为文本格式输出。

OCR 技术广泛应用于文档扫描、银行卡和身份证识别、自动识别车牌号码等领域。随着深度学习技术的发展，OCR 的准确率得到了大幅提升，未来 OCR 技术将会更加普及和应用于各种领域。

本程序使用 Python 自带的 CnOcr 库。先创建一个 Ocr 示例，然后对图像进行 `ocr.ocr(image)` 识别，返回值是一个包含 `[text, score, position]` 三元组的列表，目标是识别第三段四位数的取件码，可以根据该特征筛选识别结果，因为其他的字符都比较模糊，该方法简单且高效。

同时，由于部分取件码区域并不满足 Width 大于 Height 的设定，会有一些取件码在矫正之后变为竖着的图形。为了保证识别的准确性，进行两次 Ocr 识别第一次对旋转矫正后的图像进行 Ocr 识别，第二次对包围框内的原图像进行识别，如果第一次没有识别到四位数的取件码则进行第二次，否则不进行第二次识别。识别结果举例如图 3-20 所示。使用 Ocr 对该取件码进行识别，识别得到两个字符串，一个是“9737”，另一个是“活”，无论是根据 score 大小，还是字符串长度，都可以正确筛选出取件码信息。



图 3-20 Ocr 识别结果

3.4 信息存储

将取件码信息合理的存储到本地内存是实现快递位置可视化的重要前提。对于同一个货架同一层上的包裹，只需要快递码需要存储两个信息，一个是第三段取件码，另一个是包裹在货架中的位置，该信息可以用取件码在图像中的位置表示，也就是包裹取件码的最小外接矩形。那么，一个快递包裹的位置信息包括：第三段取件码，最小外接矩形信息（包括最小外接矩形中心坐标，最小外接矩形宽度，最小外接矩形高度，旋转角度）。在识别过程中，通过一个列表维护取件码信息，如果识别到一个新的取件码，则添加进列表，如果识别到一个已有的取件码，则更新取件码对应的最小外接矩形信息。同时，每隔 5s 将列表中的信息存储到对应的本地文件中，并且记录当前采集到的该货架图片，将图片以固定格式命名，并将图片名字存储在取件码信息文件第一行。以图 3-5 的货架快递图识别结果为例，本地文件中存储的取件码信息如图 3-21 所示。

./img/test1.png	
7863 1765.000000 561.000000 44.000000 92.000000 90.000000	
9437 1331.502197 562.424316 84.324928 52.915100 1.735705	
1348 1556.063721 561.736877 87.902275 50.918999 5.492324	
4679 1140.023560 551.126465 76.662918 50.027386 3.576334	
1 994.000000 512.000000 42.000000 52.000000 90.000000	
8888 654.542114 509.363800 45.781759 59.173558 87.207291	
4267 266.153819 508.239774 44.908574 59.040535 82.674985	
9906 430.903091 508.072593 59.462724 37.176769 12.904617	
315 144.013428 404.409363 44.111141 34.355662 8.426949	
8750 804.000000 435.000000 44.000000 40.000000 90.000000	
9499 530.400769 482.915771 42.800000 53.031864 84.987211	
3624 67.000000 478.500000 35.000000 48.000000 90.000000	
4065 1746.215594 459.941112 54.813049 97.984390 75.943740	
8011 1372.183108 447.295349 54.083275 88.419220 86.820168	
3214 1570.000000 430.000000 54.000000 94.000000 90.000000	
8080 1187.135254 422.689178 78.747116 47.475705 9.442122	
7291 266.434448 410.778748 38.749294 45.448946 84.885571	
4980 963.557739 417.424268 73.519461 48.050076 8.746141	
3611 409.528625 410.857727 41.746628 49.083187 85.426879	
8880 515.500000 392.999949 52.999992 43.999992 -0.000000	
6416 1805.785278 387.062592 100.007090 50.187545 1.287129	
6139 655.006104 376.945129 60.295620 44.062187 6.340191	
1287 131.000000 378.500000 84.000000 91.000000 0.000000	
5224 827.018372 337.164673 46.270817 42.062523 81.659811	
2886 368.000000 331.000000 42.000000 48.000000 90.000000	
1570 205.009981 321.130005 42.143570 46.527630 81.849894	
5824 1480.000000 310.999949 58.722027 93.015686 86.635545	
5436 1157.647795 307.908452 54.777069 86.726463 79.215698	
445 951.723511 284.405051 51.540054 70.867569 85.601288	

图 3-21 本地存储的取件码信息

3.5 实验结果评价

使用 Python 提供的 `time.time()` 函数分别统计取件码检测和取件码识别所用的时间，计算出的平均每件时间如表 3-1 所示，虽然取件码识别所用时间较长，但在本任务背景下是可行的。同时，统计了取件码检测和识别的召回率，分别为 100% 和 93%。

表 3-1 取件码检测和识别的召回率和所用时间

方法	识别正确快递单 数（件）	识别错误快递单 数（件）	平均识别时间（s/ 件）
取件码检测	100	0	0.0127153
取件码识别	93	7	0.3040722

第4章 可视化呈现

可视化呈现是快递领取辅助系统的关键环节，快递位置获取相当于是快递领取辅助系统的输入部分，将取件码以及位置信息存储到数据库，可视化呈现相当于是系统的输出部分，高效便捷的输出可以提高用户的快递领取体验，同时也可以提高用户的领取快递速度。主要输出功能有文字描述输出，图片标识输出和快递相对坐标输出。

4.1 软件功能实现

基于 Java 语言开发一个安卓软件^[20]，主要分为两个页面，一个是快递查询界面，可以高效便捷的输入快递单号进行查询，另一个取件码查询结果展示界面，以简洁易懂的方式给用户展示快递包裹位置。

4.1.1 需求分析和设计

需求分析和设计旨在收集用户需求，根据需求对界面、功能和流程进行设计，然后建立原型并经过测试以确保其有效性。

对于本文实现快递查询软件，使用该软件的用户需要已经使用菜鸟裹裹等软件得知了取件码，取件码是由三段数字的形式构成，第一段是货架号，第二段是描述第几层，第三段是快递编号后四位。因此，快递查询界面需要一个七位数的输入，分别对应货架号、第几层和快递编号后四位，而且需要支持修改功能，以防用户输入取件码出现错误。同时，当且仅当用户正确输入完完整的取件码之后，方可进入查询页面，否则将会出现错误跳转，故需要实现取件码是否合法。

取件码输出合法后，用户确认并跳转到快递位置展示界面，该界面需要给用户提供文字以及图片输出，对于不存在的快递应提示用户取件码无效，并返回取件码输入界面，对于存在的快递应该展示快递位置信息。

4.1.2 开发和编程

开发和编程旨在根据需求分析和设计完成的原型，在这个阶段中将根据所选择的程序语言为应用程序创建基本结构和架构。包括前端界面，后台功能和数据存储模块等。

取件码输入界面如图 4-1 所示，整体界面是一个垂直方向的线性布局，第一行是一个 TextView，内容为“快递查询”。第二行是一个网格布局，方向为水平方向，宽度比例为 2: 0.3: 1: 0.3: 4，四个网格的内容都是 TextView，这个网格布局的宽度权重主要是为了和下一行的取件码输入框对齐。第三行同样是一个水平方向的网格布局，分为 9 个网格，其中有 7 个 EditText 和 2 个 TextView，EditText 用来获取输入取件码，因为取件码总长度为 7 位，故每个 EditText 可以输入的字符类型都是 number，最大长度为 1。TextView 的内容为“-”，作为三段取件码之间的分割符。1 个 EditText 和一个 TextView 多占的宽度比例为 1: 0.3，保持和上一行的提示信息对齐。

为了保证用户输入的取件码是合法的，需要实现输入新的数字时选中下一个文本框，删除文本时焦点回退到上一个文本框，同时，不允许用户点击更换焦点文本框。所谓焦点变化，当一个 EditText 被选中时发生了两次焦点变化，先是上一个获得焦点的 EditText 失去焦点，第二个是新的被选中的 EditText 获得焦点。

首先为了实现文本框的焦点自动变更需要设置一个去全局变量 etFocus，用来记录当前焦点所在的文本框，同时对每一个文本框设置一个文本变化监听器：et_numbers.get(i).addTextChangedListener，设计 afterTextChanged 函数，当输入新的数字后，焦点位置 etFocus 加一，焦点移动到下一个文本框，当 etFocus 等于 6 且最后一个文本框也已经获得输入后，焦点停留在最后一个文本框，同时设置确认键位可用状态，否则确认键置为不可用。

然后实现用户不能修改当前选中的文本框，对每一个文本框设置一个焦点变化监听：et_numbers.get(i).setOnFocusChangeListener(this)，当发生文本框变化时，查询获得焦点的文本框与 etFocus 记录的值是否一致，如果不一致则 etFocus 记录的文本框获得焦点。

最后实现删除回退功能，设置一个键盘监听：et_numbers.get(i).setOnKeyListener()，当删除键被按下时，判断当前文本框的内容是否为空，如果为空，说明需要删除的是上一个文本框的内容，则删除上一个文本框的内容，并将焦点位置 etFocus 减 1，前一个文本框获得焦点。

如此，就实现了输入时的自动向后选中文本框和删除时焦点想前移动，同时，用户不能更改文本框的焦点位置。

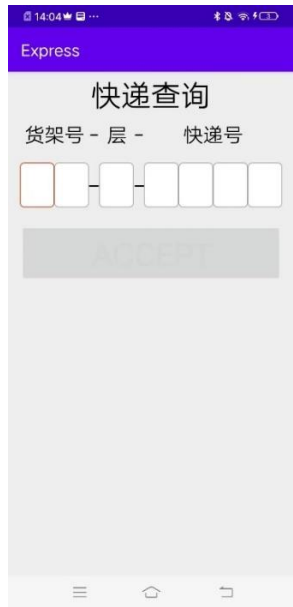


图 4-1 取件码输入界面

快递位置展示界面。首先声明一个快递实体，包含三段取件码和快递位置在图像中对应的坐标。同时，实现了一些函数比如初始化和以字符串格式输出。为了模拟访问服务器操作，在本地搭建了一个数据库，数据库表项和内容如图 4-2 所示。同时实现了数据库的创建、插入、更新、删除和查询功能。

	id	shelf	layer	number	locX	locY
1	1	07	3	8569	18.0	16.0
2	2	77	3	2444	8188.0	249.0
3	3	91	1	9476	5590.0	3911.0
4	4	05	2	3200	8322.0	9575.0
5	5	92	3	6876	3299.0	2080.0
6	6	54	4	8232	1552.0	5416.0
7	7	83	1	1333	4826.0	2706.0
8	8	60	1	6669	1988.0	7641.0
9	9	11	2	1270	2998.0	3172.0
10	10	18	3	6712	4941.0	4332.0

图 4-2 数据库

快递位置展示界面的布局，整体是一个垂直方向的线性布局，第一行包含一个返回键，使用 ImageView，剩余部分是一个 TextView，可以显示货架号信息。再往下是一个滚动视图，用来显示一个模拟的货架图片，通过在该背景图上做标记的方式展示快递在货架上的相对位置。

快递查询是一个访问数据库的过程。实现流程：用户输入取件码，点击确认时，取件码输入界面将该取件码打包，伴随事件一起传递给快递位置展示界面，快递展示界面根据取件码访问数据库。

当查询失败时，会提示用户快递不存在，用户点击确认返回取件码输入界面：



图 4-3 查询失败

当查询成功时，会弹出快递位置文字描述提示框，用户点击确定会停留在快递位置展示界面，快递位置展示图像是根据数据库中记录的快递位置坐标，在货架背景图像上使用红色圆点代表快递位置。用户可以按返回键或则左上角的返回按钮返回取件码输入界面。查询成功结果如图 4-4 所示：

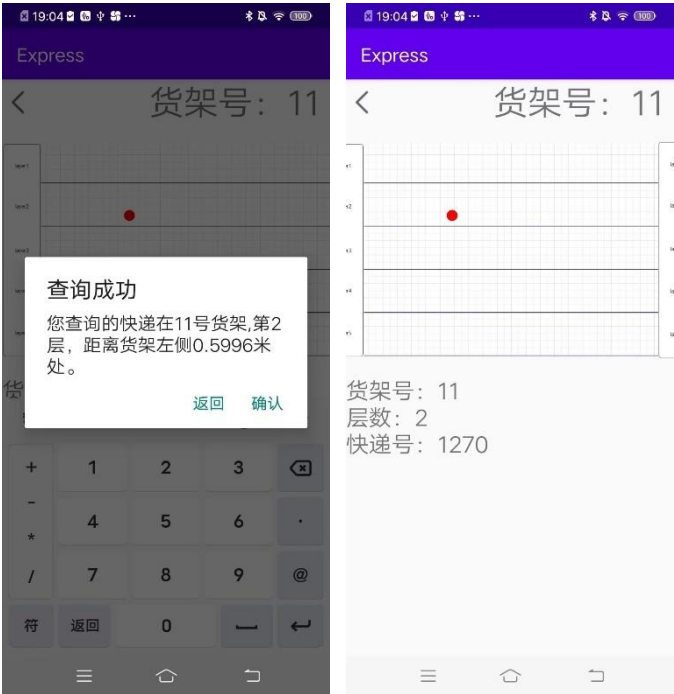


图 4-4 查询成功

4.2 其他输出方式

由于在取件码检测过程中保存了拍摄的真实货架图像和取件码对应的最小外接矩形的一系列信息，还可以在原图中标记出取件码对应的最小矩形框。由于本实验没有搭建服务器，故手机软件无法访问电脑端存储的数据，故该功能使用 python 实现，基于手机软件的实现该功能只是多了一个访问服务器步骤，其他基本类似。

和手机输入取件码类似，用户输入取件码，如图 4-5 所示。

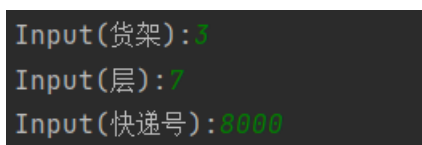


图 4-5 取件码输入

然后根据取件码的货架号和层号确定将要访问的本地文件，将本地文件中的每一行读入一个列表，列表第一行元素为对应的货架图片名字，将第一行读出，得到图片的存储路径，然后将剩余数据按行读出并令牌化，根据第三段取件逐一比较。

如过取件码查询失败，则输出不存在该快递。如果查询成功，则在对应图片中使用最小矩形框出取件码所在位置，如图 4-6 所示。

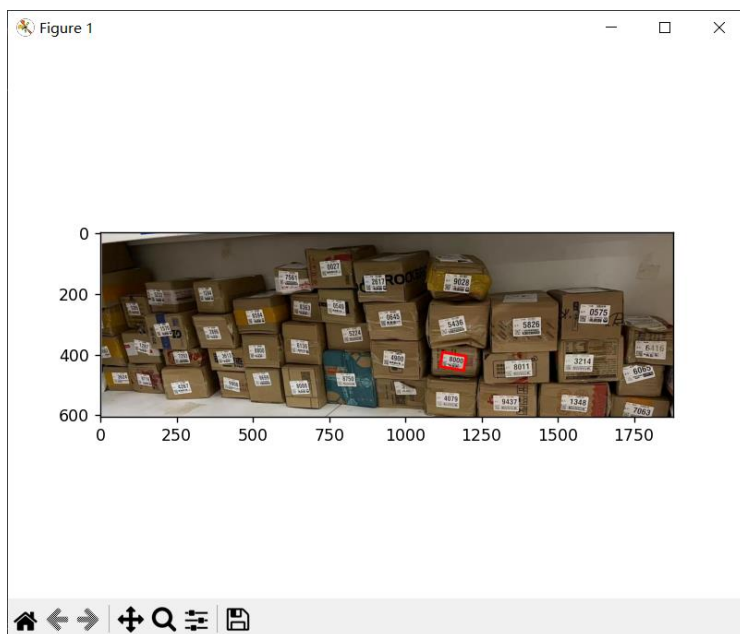


图 4-6 取件码位置展示

4.3 本章小结

以上就是实现的 3 种快递位置展示方式，分别是文字输出、相对位置输出和图像输出，三种方式结合起来能够给用户提供清晰的快递位置描述。在实验的开始，还设想了一种其他的输出方式，指示灯提示和激光指示，但由于这两种设备不具备并行功能，即同一时间段只能服务一个用户，不满足加速快递领取的要求，故没能实现。

第5章 结束语

5.1 工作总结

本文通过分析学校、小区等大型快递领取中心的快递领取现状，发现存在的用户寻找快递慢导致的快递领取效率低下，快递驿站取快递人员拥挤等问题，设计了计算机视觉辅助的快递领取辅助系统。该系统基于已有的计算机视觉技术检测并识别快递单号，并通过手机软件的形式为取快递用户提供简洁易懂的快递位置可视化。本文的主要工作内容如下：

1. 通过实现快递分拣任务常用的检测方法（特征匹配方法），发现该方法并不适用于多目标检测，提出基于双阈值 OTSU 阈值化和边缘检测等检测取件码位置的方法，然后又使用了图像矫正和光学字符识别的方法，实现了快递放入货架过程中的快递位置信息记录。

2. 通过安卓软件和 Python 程序的方法为取件用户提供了多种快递位置可视化方法：文字描述，相对位置描述，图片输出。三者结合能够清楚的展示快递的位置，同时简介高效的取件码输入界面和清楚明白的快递位置展示界面可以提高用户体验。

3. 开发过程中，基于收集的菜鸟驿站快递包裹和实验室的环境搭建了一个简易的货架场景，如图 5-1 所示，验证了本系统的稳定性和结果的准确性。



图 5-1 模拟货架场景

5.2 工作展望

本文在对快递领取任务进行深入分析的基础上，基于已有计算机视觉方法和软件开发方法实现了快递领取辅助系统，其功能和效率基本满足预期要求。但在本文的基础上，该系统仍存在优化空间：

1. 在区间码检测方面，如果能够收集足够的数据集，可以使用深度学习的目标检测模型，比如 YOLO^[21]等现常用模型进行训练，应该可以提高取件码检测的准确率。同时，可以通过加入深度的方式，在信息采集过程中，采集快递包裹的三维信息，更加丰富的快递信息可以为用户提供更好的可视化展示。

2. 在软件开发方面，可以使用更加美化和简洁的界面为用户提供更良好的体验，此外可以通过结合三维的快递位置信息，实现快递在三维空间的展示，还可以结合虚拟现实的方式提升用户体验。

3. 目前本系统只能解决较为简单的快递摆放场景，对于研究背景中提及的复杂快递摆放场景仍没有办法处理，主要是由于快递位置获取的速度和精度不够，如果结合一个深度学习模型，有望得到解决。

致 谢

首先，我要感谢我的指导教师赵海森教授。在整个毕业论文的完成过程中，老师给予了我严谨、耐心、细致的指导和帮助，为我的毕业论文提出了许多宝贵的建议。感谢老师对我的支持和鼓励，使我在论文撰写过程中能够克服难题，不断前进。

其次，我想感谢我的母亲。在我整个大学的学习和生活中，我的母亲一直是我最强有力的后盾。她的支持和鼓励是我一路前行的动力，让我有勇气和信心突破自己的极限。我感到自己很幸运，有这么一位伟大的母亲。

最后，我要感谢我的朋友们。感谢他们陪我度过了这段美好的大学时光。在生活和学业的各个方面，您们给了我很多帮助和鼓励，是我在大学期间最好的伙伴和支持者。

值此论文完成之际，再次感谢我的指导教师赵海森教授、母亲和朋友们的支持和帮助。感谢他们在我人生道路上提供了无私的帮助和鼓励，我会一如既往地珍惜这份感情，勇往直前，努力奋斗！

参考文献

- [1] 张守龙.浅析电子商务对快递行业发展的影响[J].中国商论,2022(15):37-39.DOI:10.19699/j.cnki.issn2096-0298.2022.15.037.
- [2] 王锦凯,宋锡瑾.计算机视觉技术应用研究综述[J].计算机时代,2022,No.364(10):1-4+8.DOI:10.16644/j.cnki.cn33-1094/tp.2022.10.001.
- [3] 张锦,陈义友.物流“最后一公里”问题研究综述[J].中国流通经济,2015,29(04):23-32.DOI:10.14089/j.cnki.cn11-3664/f.2015.04.004.
- [4] 赖丽婷,周紫婷,王玲等.校园快递领取智能系统研究[J].电脑知识与技术,2019,15(20):218-221.DOI:10.14004/j.cnki.ckt.2019.2268.
- [5] Yunhao Yang. Design and Implementation of Campus Express Delivery on WeChat Public Platform[J]. Academic Journal of Computing & Information Science,2022,5(11).
- [6] 王雅平,吕婧冉.基于智能快递柜末端配送问题研究[J].办公自动化,2022,27(23):26-28+32.
- [7] 李薇薇,于博.机器视觉检测系统的组成[J].信息系统工,2018,No.289(01):104.
- [8] 倪桥. 基于机器视觉的自动化分拣系统的设计与研究[D].安徽理工大学,2020.DOI:10.26918/d.cnki.ghngc.2020.000518.
- [9] 秦蓉.基于 Java 语言的安卓软件开发研究[J].电子技术与软件程,2021(05):44-45.
- [10] Arshad Saba,Kim Gon Woo. A Robust Feature Matching Strategy for Fast and Effective Visual Place Recognition in Challenging Environmental Conditions[J]. International Journal of Control, Automation and Systems,2023,21(3).
- [11] Sonali S. Thangan,Ankit R. Mune. Review on Distinctive Image Features from Scale-Invariant Key-Points[J]. International Journal of Soft Computing and Engineering (IJSCE),2018,8(2).
- [12] GU ZHIWEI,SCOTT MATTHEW R,CHEN GANG,TIEN JONATHAN Y. QR CODE DETECTION[P]. : US2011290882,2011-12-01.
- [13] 谭光兴,张伦.基于改进 SIFT 的图像特征匹配算法[J].广西科技大学学报,2022,33(02):41-47.DOI:10.16375/j.cnki.cn45-1395/t.2022.02.006.
- [14] 宋森森,贾振红,杨杰,Nikola KASABOV.结合 Ostu 阈值法的最小生成树图像分割算法[J].计算机工程与应用,2019,55(09):178-183.
- [15] 冈萨雷斯.数字图像处理[M].欸经电子工业出版社, 2007.
- [16] 肖扬,周军.图像边缘检测综述[J].计算机工程与应用,2023,59(05):40-54.
- [17] Kittler J. On the accuracy of the Sobel edge detector[J]. Image and Vision Computing,1983,1(1).
- [18] Tsoi Y.-C.,Brown M.S.. Multi-view document rectification using boundary[J]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2007.
- [19] 王栋.人工智能 OCR 技术的应用研究[J].电子技术与软件工程,2022,No.219(01):122-125.
- [20] 罗娟. 基于 Android 的智能家居控制端软件设计[D].中国科学技术大学,2018.
- [21] Guo Hao,Yang Jiahua. An improved object detection algorithm based on YOLO*[J]. Journal of Physics: Conference Series,2022,2216(1).