

# 第十五中国研究生电子设计竞赛

## 技术论文

人机协同环境下基于行为克隆的云基机械臂布料堆叠

Cloud-based robotic arm cloth stacking based on  
behavioral cloning in human-machine collaborative  
environment

参赛单位： 山东大学控制科学与工程学院

队伍名称： 周が一番強いです

指导老师： 周风余、宋锐

参赛队员： 顾潘龙，付天宇，延晓晖

完成时间： 2019.6.19

## 目 录

摘要 .....	3
Abstract .....	4
第一章 作品难点与创新 .....	5
1.1 作品难点 .....	5
1.2 作品创新点 .....	5
1.2.1 动态场景下的工作台定位 .....	6
1.2.2 基于 Blend Mask 的行为克隆 .....	6
1.2.3 基于强化学习的布料码垛实现 .....	6
1.2.4 算法在云平台的部署 .....	7
第二章 方法论证与设计 .....	8
2.1 方法论证 .....	8
2.2.1 图像分割方案论证 .....	8
2.2.2 基于强化学习机械臂布料码垛的方案论证 .....	9
2.2 方案设计 .....	9
第三章 原理分析与硬件组成 .....	11
3.1 原理分析 .....	11
3.1.1 Apriltag 标签识别 .....	11
3.1.2 基于 Blend Mask 的行为克隆 .....	14
3.1.3 基于深度强化学习的布料码垛实现 .....	17
3.2 硬件组成 .....	20
3.2.1 摄像头 .....	20
3.2.2 机械臂 .....	21
3.2.3 机械臂端的工控机 .....	22
3.2.3 云服务器 .....	22
第四章 软件设计与流程 .....	23
4.1 整体流程设计 .....	23
4.2 图像处理流程设计 .....	24

4.3 强化学习码垛流程设计 .....	25
<b>第五章 系统测试与误差分析 .....</b>	<b>27</b>
5.1. 系统测试 .....	27
5.1.1 布料堆叠 .....	27
5.1.1 行为克隆 .....	28
5.2. 基于 AprilTag 码的仿射变换实验误差分析 .....	29
5.3 机械臂移动实验误差分析 .....	30
5.4 强化学习布料堆垛学习误差分析 .....	32
5.5 基于 BlendMask 的观察学习误差分析 .....	33
5.5.1 图像标注误差 .....	33
5.5.2 模型误差 .....	33
<b>总结 .....</b>	<b>34</b>
<b>参考文献 .....</b>	<b>35</b>

## 摘要

本次设计通过识别工作台上 Apriltag 标签的位置，将工作台区域仿射变换到像素坐标系的固定位置。然后使用实例分割获取图像中所有实例的位置、尺度、偏航角等信息，通过深度强化学习完成布料的堆垛。同时，本设计基于实例分割设计了行为克隆算法，在时间上对人类行为改变的环境信息进行编码，并将其传输到机械臂控制工控机对此人类行为进行复现。同时，本设计所有对 GPU 算力有较高要求的算法部署在实验室的云服务器上，以此实现对图像分割、行为克隆等算法的实时运算。

通过上述设计，本设计满足了纺织工业人机协作中对固定手眼系统中相机移动的需求。完成了对散乱堆叠布料的实时检测和堆叠整理的工作。实现了对人类行为的克隆，完成机械臂动作组的自动编码，缩减了生产线中布料堆叠任务更换，所导致的机械臂动作组的重新编码对时间的浪费。

本文中的主要创新点有四，本设计借助 Apriltag 和仿射变换令机械臂固定手眼系统中的相机可以自由移动；本设计首次在散乱布料的识别中引入实例分割，以此识别布料实例的位置与朝向角，同时设计了行为克隆算法，实现了对机械臂布料堆叠的自动编码，与行为动作复现；本设计引用深度强化学习从视觉角度上完成布料堆垛，相比传统堆垛方法本设计增加了堆垛精度；本设计将所有需要消耗大量 GPU 运算资源的算法部署在云服务平台，以此实现图像的实时处理。

**关键词：仿射变换，实例分割，行为克隆，云服务器**

## Abstract

In this design, by identifying the position of apriltag label on the workbench, the workbench area is affinely transformed to the fixed position of pixel coordinate system. Then, the position, scale, yaw angle and other information of all instances in the image are obtained by instance segmentation, and the stacking of cloth is completed by deep reinforcement learning. At the same time, this design designs a behavior clone algorithm based on instance segmentation, encodes the environmental information of human behavior change in time, and transmits it to the manipulator to control the industrial computer to reproduce the human behavior. At the same time, all algorithms which have high requirements for GPU computing power are deployed on the cloud server in the laboratory, so as to realize the real-time operation of image segmentation, behavior cloning and other algorithms.

Through the above design, this design meets the needs of camera movement in the fixed hand eye system in human-computer cooperation of textile industry. The real-time detection and stacking finishing of the scattered stacked fabrics are completed. It realizes the cloning of human behavior, completes the automatic coding of manipulator action group, and reduces the waste of time caused by the change of cloth stacking task in the production line.

There are four main innovations in this paper. With the help of apriltag and affine transformation, the camera in the fixed hand eye system of the manipulator can move freely. For the first time, the case segmentation is introduced in the identification of scattered cloth, so as to identify the position and orientation angle of cloth instance. At the same time, the behavior clone algorithm is designed to realize the automatic coding and behavior movement of the cloth stacking of the manipulator Compared with the traditional stacking method, this design increases the stacking accuracy; this design deploys all the algorithms that need to consume a lot of GPU computing resources on the cloud service platform, so as to realize the real-time image processing.

**Keyword: Affine transformation, instance segmentation, behavior clone, cloud server**

## 第一章 作品难点与创新

### 1.1 作品难点

本次赛题选择为人机协同环境下基于行为克隆的云基机械臂布料堆叠，主要应用在纺织工业。

纺织生产过程中，每一次更换纺织任务都需要重新编程机械臂抓取流程以适应不同的布料堆叠顺序及形状。对机械臂的重新编程会消耗大量的人力、时间成本，减缓生产速度，且在布料属于非刚性物体，无法使用物体检测算法直接获取其位置与姿态。针对该问题，本文设计了基于图像实例分割的行为克隆算法，通过机械臂模仿人手堆叠布料的过程，适应环境完成对新加入纺织任务的自动编程。在此过程中面临的主要难点如下几方面：

- 1、纺织任务中难免需要进行人机协同任务，对相机的固定方式及摆放位置有着严苛的要求，且需要相机进行移动，以获取尽可能大的识别区域；
- 2、布料属于非刚体，无法通过物体检测算法获取布料实例的位姿，且行为克隆算法需要较为精确的布料检测算法及位置感知系统的加持；
- 3、工业机器人对缝制作业过程中动态多变的环境参数、随机扰动等因素适应性较差，且常规工业机器人作业工艺的开发与调试周期长、效率低、工作量大，灵活度低、算法适应性差。
- 4、行为克隆、实例分割需要强力 GPU 算力的服务器加持。

### 1.2 作品创新点

本设计针对上述设计难点展开，在 Blend Mask 实例分割算法的基础上加入面向人类行为的感知，对由人类行为产生的工作台环境变化进行编码，完成行为克隆算法的开发。进一步的，通过 Apriltag 快速视觉标签提取工作台所在平面，以减小相机位置变化对布料位置的检测精度与机械臂抓取精度产生的影响。进一步的，通过深度强化学习搭建机械臂布料码垛技能学习框架，实现机械臂端到端的布料码垛运动控制。最后将算法部署在云服务器，能够同时完成对多台机械臂

本地的调度与部署，解决算法对 GPU 算力的需求，降低了本地机械臂对工控机性能的需求，有利于后续本地设备成本的进一步缩减。

### 1.2.1 动态场景下的工作台定位

本文设计在机械臂工作平台的四角黏贴 Apriltag 标签，通过对标签的识别和四个标签的联立误差补偿，获取工作台四角在空间中的真实坐标。并在工作台一角出现遮挡时使用未被遮挡的三个角估算被遮挡的角的真实坐标。最后应用仿射变换将工作台四角变换到相机坐标系中固定的四个点，获取布料实例和机械臂在工作台平面的绝对位置。

通过此工作，可以将工作台平面固定显示在像素坐标系设定区域中，为图像识别任务铺垫了数据基础，减小了固定手眼系统中由于相机位置变化所引入的识别误差。

并且，此工作改变了传统手眼系统需要相机固定约束条件，可以将相机部署在工作台相机视野可视的任意区域，为布料堆叠后，人机协作的纺织工作提供了便利。

### 1.2.2 基于 Blend Mask 的行为克隆

本设计在 Blend Mask 图像分割算法的基础上进行，通过对机械臂工作环境的语义信息进行编码，获取到人类的行为信息，进而保证机械臂可以根据此信息克隆人类的行为，完成预设的任务。

该设计首次将实例分割和行为克隆算法应用在布料识别领域，获取布料坐标信息进行布料抓取，并通过分析人类行为序列获取动作信息，实现简单快捷的学习任务。通过简单的人为操作布料堆叠，能够轻松将堆叠算法部署到实际的工作场景中。

### 1.2.3 基于强化学习的布料码垛实现

本设计针对传统常规方法设计的机器人作业不能动态地适应非结构化工作环境或场景多变的工作场合问题，基于环境图像数据，利用深度强化学习算法，

在马尔可夫过程中引入 Q 学习策略机制，构建深度策略网络，实现由数据到动作的端到端机器人技能的训练，实现视觉感知角度的上的布料堆叠。

#### 1.2.4 算法在云平台的部署

深度学习在计算机视觉领域已取得了很大的成功，强化学习也越来越多的被用于机器人运动控制。针对问题特点，本次作业采用了诸如 Blend Mask 等需要大量 GPU 运算的模型。为保证模型的识别精度，并为模型加入新类型的布料数据，需要采集大量的布料堆放图片作为训练数据，随时为模型提供新的训练数据，则更需要云服务器的加持和云平台的部署与调度。

本次作业的执行单元使用 UR 5e 系列机械臂，它具有可用于远程访问控制的以太网接口，支持控制 I/O 编程。图像实体分割模型的输出作为机械臂控制程序的输入，控制机械臂运动完成移动和抓取动作。

为了简化模型创建，实现上述功能，提高训练速度，方便模型评估，本次作业依托实验室云服务平台完成网络模型训练和图像实例分割以及机器臂控制任务。这样不仅能充分利用云平台带来的算力提升，另一方面这种策略也使得实验设备可以跨局域网调试运行，大大简化了实验环境，方便实验过程。



## 第二章方法论证与设计

### 2.1 方法论证

#### 2.2.1 图像分割方案论证

图像分割是计算机视觉研究中的一个经典难题,已经成为图像理解领域关注的一个热点,图像分割是图像分析的第一步,是计算机视觉的基础,是图像理解的重要组成部分,同时也是图像处理中最困难的问题之一。所谓图像分割是指根据灰度、彩色、空间纹理、几何形状等特征把图像划分成若干个互不相交的区域,使得这些特征在同一区域内表现出一致性或相似性,而在不同区域间表现出明显的不同。简单的说就是在一副图像中,把目标从背景中分离出来。对于灰度图像来说,区域内部的像素一般具有灰度相似性,而在区域的边界上一般具有灰度不连续性。

关于图像分割技术,由于问题本身的重要性和困难性,从 20 世纪 70 年代起图像分割问题就吸引了很多研究人员为之付出了巨大的努力。虽然到目前为止,还不存在一个通用的完美的图像分割的方法,但是对于图像分割的一般性规律则基本上已经达成的共识,已经产生了相当多的研究成果和方法。

##### 1. 传统分割方法

这类方法是利用数字图像处理、拓扑学、数学等方面的知识进行图像分割。主要包括:基于阈值的分割方法、基于区域的图像分割方法、分水岭算法、基于边缘检测的分割方法以及结合特定工具的图像分割算法等。这一部分的优点为计算简单、效率较高、边缘定位准确,缺点分割效果和精度无法与基于深度学习的方法相比较<sup>[1]</sup>。

##### 2. 基于深度学习的分割算法

深度学习热度居高不下,许多基于神经网络的分割算法也相应提出,这些算法几乎涉及到了深度学习的所有方面,包括:卷积神经网络(CNNs)、循环神经网络(RNN)、编码解码框架(Encoder-Decoder)、对抗神经网络(GANs)、迁移学习等<sup>[17]</sup>,在这些相关知识的基础上,专家学者提出了多种精度高、效果好的图像分割

模型，如：FCIS、Mask R-CNN、YOLACT、Polar Mask、SOLO、Point Rend、Blend Mask 等<sup>[18]</sup>。基于深度学习的算法优点为：精度高、效果好，但其对于应用设备的硬件要求较高，需要保证合适的 GPU 环境<sup>[2]</sup>。

考虑到本项目中，实际场景较为复杂，对识别算法的准确度有一定要求，因此采用基于深度学习的分割算法，且在考察相关分割算法后，使用精度和速度均在最优水平的 Blend Mask 算法<sup>[3]</sup>作为基础算法。

### 2.2.2 基于强化学习机械臂布料码垛的方案论证

目前常用机器人布料码垛方案主要有基于机器人示教编程，基于视觉引导，基于技能学习等方式<sup>[19]</sup>。几种方法大体思想如下。

#### 1. 示教编程

由人工导引机器人末端执行器(安装于机器人关节结构末端的夹持器)，或由人工操作导引机械模拟装置，或用示教器来使机器人完成预期的动作<sup>[4]</sup>

#### 2. 视觉引导

机器人视觉引导是将机器视觉与运动控制相结合，利用视觉系统获取分析图像，定位所要抓取物体在图像中的位置。通过标定将图像中的位置数据转换为运动控制系统的坐标中，实现机械人定位抓取动作<sup>[5]</sup>。

#### 3. 强化学习

强化学习(Reinforcement Learning, RL)是用于解决智能体(agent)在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的算法。本文针对以上三种方案各自的优势和特点，从灵活性、应用背景、机器人技能提升等方面综合考虑，最终选择第三种方案，即强化学习。该方案可以使机器人在复杂、动态的环境中学习并获取作业能力，并根据目标约束条件不断在学习中进行优化，可以有效提高工业生产效率<sup>[6]</sup>

## 2.2 方案设计

综上所述，本设计将人机协同环境下基于行为克隆的云基机械臂布料堆叠划分为四个任务。

### 1. 工作台区域的俯视图变换

为了确保相机的移动不会对识别得到的物体位置产生影响。本文使用 Apriltag 标签获取工作台的真实位置，并通过仿射变换将工作台所在区域变换到像素坐标系下的固定位置。

### 2. 行为克隆

在该部分，实例分割算法获取工作台上散乱摆放的布料的实例范围，通过求解实例 mask 的最小包围框输出其中心点、尺寸及朝向，在时间序列上编码人类对工作台上摆放布料的影响。然后将该编码送入机械臂完成人类动作的复现。

### 3. 机械臂布料码垛技能学习

为使机械臂学会布料码垛技能，本文通过使用深度强化学习算法，搭建机械臂布料码垛技能学习框架，采用端到端的控制方式、将图像输入转变为机械臂末端执行动作，从而获得机械臂布料码垛作业能力。

### 4. 云平台部署

本次作业的数据流涵盖相机数据采集，图像实体分割/强化学习动作模仿，图像实体位姿变换，机械臂控制四个环节。其中深度学习预测模型，位姿变换和机械控制都在云端完成。云端各种服务间采用分布式内存数据库作为消息总线，完成服务分发功能。

图像采集的图像和视频数据，通过 HTTP 协议上传到云端数据采集服务进程。HTTP 是基于文本的解析协议，天然支持自定义命令，完成协议扩展。机械臂控制信息的下发，则通过机械臂 SDK 下发到终端。

以布料抓取过程为例，数据采集服务根据需要将数据写入数据总线，并触发下游任务，比如体分割的图像数据写入 Blend Mask 模型进程，触发实体分割任务；分割完成后，各实体的识别数据再次推送到数据总线；实体位姿变换服务获取到实体数据后，将实体数据转换成机械臂控制信息；机械臂控制服务从数据总线获取到控制信息后，进一步将其转化为控制信号，通过机械臂控制 SDK 实现机械臂远程控制。

以上，实现了相机图像信息控制机械臂移动和抓取的过程。在以上过程中，大量工作都在云端完成。

## 第三章原理分析与硬件组成

### 3.1 原理分析

#### 3.1.1 Apriltag 标签识别

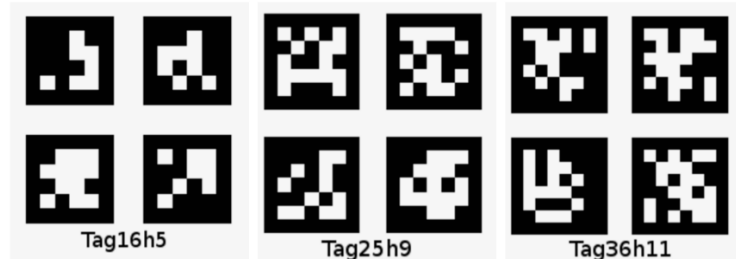


图 3-1 AprilTag 码

如图 3-1 所示，AprilTag 是由密西根大学 April 实验室开发的一个鲁棒灵活的基准系统<sup>错误!未找到引用源。</sup>。该算法有明确的目标和应用，相较于传统的视觉地标识别系统，增加了对图标的识别编码，具有更好的鲁棒性。并且在快速识别的同时能够有效处理遮挡、翘曲和透镜变形。基于 AprilTag 定位主要包含已下三个步骤。

##### [1]边缘识别

通过 Canny 算子<sup>[8]</sup>，寻找图像中的边缘。Canny 边缘检测算法首先使用高斯核对图像的各个像素点进行卷积，达到平滑图像和抑制噪声的作用；用一阶偏导的差商来计算梯度的幅值和方向，根据得出的幅值对图像的线条进行提取；对梯度幅值进行非极大值抑制，使得边缘的像素点达到最佳值；用双阈值算法检测和连接边缘，将边缘二值化，并形成边缘轮廓。具体过程如下：

首先，使用高斯滤波对图像进行平滑操作，设  $S(x, y)$  代表卷积处理后的图像， $G(x, y)$  代表二维高斯函数，如式 (3.1) 所示。 $f(x, y)$  输入原始图像。他们之间的关系如式子 (3.2) 所示。

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.1)$$

$$S(x, y) = G(x, y) * f(x, y) \quad (3.2)$$

式(3.2)中\*代表卷积符号,  $\sigma$  代表标准差, 反应平滑程度。然后使用一阶导数的有限差分进行平滑后图像的梯度值和方向的计算。梯度幅值的计算公式(2个方向)为

$$G_x = (f[i+1, j] - f[i, j] + f[i+1, j+1] - f[i, j+1]) / 2 \quad (3.3)$$

$$G_y = (f[i, j] - f[i, j+1] + f[i+1, j] - f[i+1, j+1]) / 2 \quad (3.4)$$

$$M[i, j] = \sqrt{G_x^2 + G_y^2} \quad (3.5)$$

式(3.3)和(3.4)中:  $f(i, j)$  为  $(i, j)$  点的图像灰度值。 $G_x$  为 X 方向的梯度幅值;  
 $G_y$  为 Y 方向的梯度幅值。式(3.5)中 M 为  $(i, j)$  点的幅值

接着利用非极大值抑制对梯度增幅值进行处理, 利用该梯度方向上每个像素点的梯度增幅值, 判断是否是局部的极大值。如果该像素点的梯度增幅值不是极大值点, 使用 0 来表示它的灰度值, 它将不会作为边缘点保留下来; 如果该像素点的梯度增幅值是极大值点, 则将它作为边缘点保留下来。

最后使用双阈值进行边缘的检测, 假设高低阈值为  $T_H, T_L$  图像中的每个像素点, 利用高低阈值判断是否保留边缘图像中的边缘像素点。假设图像中的一个像素点的梯度增幅值高于  $T_H$ , 该像素点被保留; 低于  $T_L$  该像素点被去除掉; 介于高低阈值之间的, 需要判断该像素点的 8 领域上的空间像素和  $T_H$  的关系, 如果高于, 则被保留。

## [2] 四边形检测

接着, 算法运用边缘结构分析(Topological structural analysis of digitized binary images)<sup>[16]</sup>确定图像中的多边形及其嵌套关系, 并使用该嵌套关系和多边形的尺度位置信息编码原图像。错误!未找到引用源。

针对该过程, 算法采用编码的思想, 赋予不同的边缘以不同的编码值, 确认多边形的层次关系。从图像起点开始, 编辑边缘的像素, 寻找与起始点同类的边缘点, 当再一次扫描至起始点时, 确认多边形的闭环形成。然后切换至下一个起始点重复该操作, 直到所有的二值点都被遍历。

最后删除掉所有边数小于 4 的多边形, 应用多边形凸包寻找算法(Finding the

convex hull of a simple polygon)<sup>[15]</sup>，计算每一个多边形本身的凸包，并求凸包及其多边形的面积，删除掉多边形面积与凸包面积的比值超出阈值的多边形，并使用该四边形的四个顶点对该四边形进行表示。

通过该算法可以有效的排除非凸多边形，四边形边框的检测结果如图 3-2 所示：

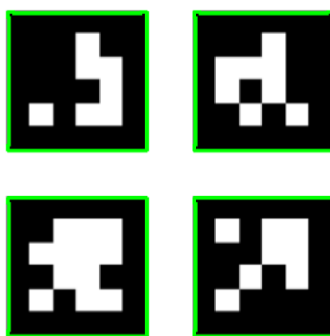


图 3-2 AprilTag 四边形检测

### [3] 编码与解码

在这一步，算法对上一步得到的四边形进一步编码完成匹配工作。

通过四边形检测过程，可以从输入图像中检测到图像中的四边形，接下来的工作就是要解析图像中所有四边形内部是否是 Apriltag 的标识码。Apriltag 标识码可以分为内部 6x6 区域的有效载荷以及最外层一圈的黑色区域两部分，编码示意如图 3-3 所示：

编码过程的伪代码展示如下<sup>[20]</sup>：

<b>输入</b>	四边形 Quad，灰度图像 Gray
<b>输出</b>	编码结果 Tagcode
1.	<b>function</b> DECONDEQUAD(Quad, Gray)
2.	Tagcode $\leftarrow 0$
3.	PointsIndex $\leftarrow 0$ 取得四边形顶点(Quad)
4.	BlackValue $\leftarrow$ 计算得黑色块的灰度均值(PointIndex, Gray)
5.	WhiltValue $\leftarrow$ 计算得白色块的灰度均值(PointIndex, Gray)
6.	Threshold $\leftarrow$ 计算出阈值(BlackValue, WhiteValue)
7.	<b>for</b> Point in PointsIndex <b>do</b>
8.	Tagcode 左移一位
9.	<b>If</b> Point 处的灰度值>Threshold <b>then</b>
10.	Tagcode 与1进行运算
11.	<b>end if</b>
12.	<b>end for</b>
13.	<b>retur</b> Tagcode

#### 14. end function

因此，在检测到外圈黑色外框之后，需要检测内部是否具有多个矩形，如果是则进行解码操作，通过提取角点坐标，将它们映射为正方形，与 Tags 库进行匹配，判断是否属于 Tags 库，如果是，则返回相应 tag 的 ID 和 tag 在图像中四个角点的坐标。如果不是，则认为检测到的四边形不是已有的 Apriltag 标识码。

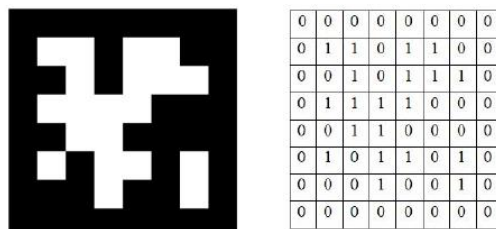


图 3-3 Apriltag 编码示意图

### 3.1.2 基于 Blend Mask 的行为克隆

#### 3.1.2.1 Blend Mask 分割算法原理分析

Blend Mask 算法是如今效果最好的且速度最快的分割算法，其包含检测网络和分掩码分支两个分支，掩码分支包含 3 个部分，一个预测得分图的底部模块，一个预测实例注意力的顶部模块，以及一个将得分和注意力融合起来的 Blend 模块<sup>[9]</sup>。其网络模型如图 3-4 所示。

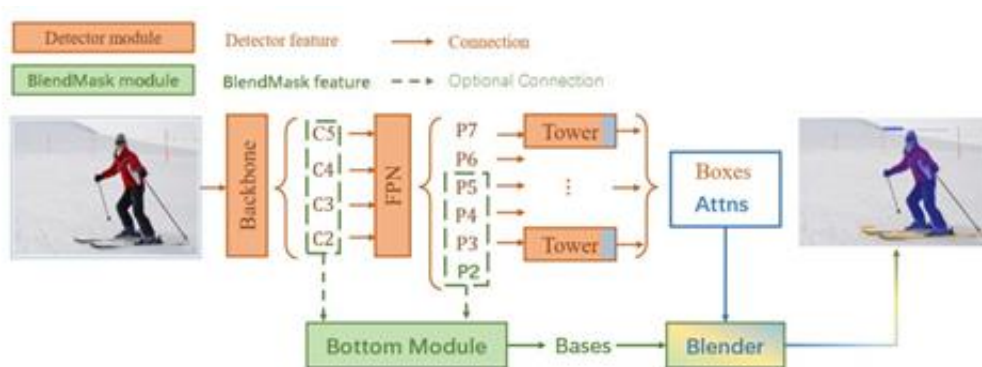


图 3-4 Blend Mask 模型图

#### 1) 底部模块

与其它基于候选框的全卷积网络类似，作者增加了一个底部模块来预测得分

图，称之为 bases (B)。B 为形状大小  $N \times K \times \frac{H}{s} \times \frac{W}{s}$  前向网络生成的特征图。N 是

batch size,  $K$  是 bases 的个数,  $H \times W$  是前向特征图输入尺寸,  $s$  是得分图输出的步长。Blend Mask 使用 Deep Lab V3+ 作为 decoder。底部模块的输入作为主干网络特征, 与传统的语义分割网络, 诸如 YOLACT 和全景 FPN 中的特征金字塔一样。

## 2) 顶部模块

Blend Mask 在每个检测模块之上都附加了一个卷积层, 用于预测注意力权重  $A$ 。这个注意力权重与 YOLACT 中的掩码系数不同, 掩码稀疏在每个分辨率为  $W_l \times H_l$  的特征图中, 输入的形狀为  $N \times K \times W_l \times H_l$ , 而 Blend Mask 的  $A$  则是一个张量, 形状为  $N \times (K \cdot M \cdot M) \times W_l \times H_l$ ,  $M \times M$  是注意力图的分辨率。

基于此三维结构, 注意力图就可以对实例级别信息进行编码, 如物体的形状和姿态。在自上而下的方法中,  $M$  通常要比掩码预测小, 因为只需要一个近似的估计值, Blend Mask 中, 使用一个输出通道数为  $K \cdot M \cdot M$  的卷积来预测。在将它们传入下一个模块之前, 首先要进行 FCOS 的后处理操作, 选取前  $D$  个边框的概率值  $P$ , 以及它们对应的注意力图  $A$ :

$$P = \{p_d \in \mathbb{R}_{\geq 0}^4 \mid d = 1, \dots, D\} \quad (3.6)$$

$$A = \{a_d \in \mathbb{R}^{K \times M \times M} \mid d = 1, \dots, D\} \quad (3.7)$$

## 3) Blender 模块

对于该模块, 首先明确 Blender 模块的输入:

- 1) 检测金字塔生成的 bbox proposal( $P$ ), 维度为  $(K \times H' \times W')$ ; 此外, 在训练的时候, 直接使用 GT bbox 作为  $P$ , 而在推理时, 则使用检测器的检测结果。
- 2) 顶层模块生成的 top-level attention( $A$ ), 维度为  $(K \times M \times M)$
- 3) 底层模块生成的 base( $B$ ), 是整图大小的  $k$  个 mask, 维度为  $(K \times H \times W)$

对于  $B$  使用 Mask R-CNN 中的 RoIPooler (即 sampling ratio 为 1 的 RoIAlign, Mask R-CNN 中为 4), 在  $B$  上裁剪出  $P$  对应的 mask, 并重置大小到  $R \times R$  大小的特征图, 最后得到的  $r_d$  的维度为  $(K \times R \times R)$ ;



$$r_d = \text{ROI Pool}_{R \times R}(B, p_d), \forall d \in \{1, \dots, D\} \quad (3.8)$$

对于  $A$ ，作者根据 FCOS 中的后处理方法，选出前  $D$  个检测框和对应的  $A$ ，并通过 RoIAlign，将  $A$  的维度由  $(K \times M \times M, H', W')$  调整为  $(K \times M \times M)$ ，记为  $a$ ，由于  $M$  一般小于  $R$ ，做一个插值，将  $A$  从  $M \times M$  插成  $R \times R$  大小，得到的  $a_d$  维度为  $(K \times R \times R)$ ：

$$a_d' = \text{interpolate}_{M \times M \rightarrow R \times R}(a_d) \quad (3.9)$$

然后在  $K$  维度上做 softmax，得到一系列的特征得分图， $s_d$  的维度同样为  $(K \times R \times R)$ ：

$$s_d = \text{softmax}(a') \quad (3.10)$$

最后把  $k$  个 bbox 大小的 mask 和对应的 attention 乘起来，再按通道叠加起来，得到最终的 mask：

$$m_d = \sum_{k=1}^K s_d^k \circ r_d^k \quad (3.11)$$

#### 4) BlendMask 的优势：

- 1) 计算量小：使用一阶段检测器 FCOS，相比 Mask R-CNN 使用的 RPN，省下了对 position-sensitive feature map 及 mask feature 的计算，
- 2) 提出 **attention guided blender** 模块来计算全局特征，相比 FCN 和 FCIS 中使用的较复杂的 hard alignment 在相同分辨率的条件下，减少了十倍的计算量
- 3) mask 质量更高：BlendMask 属于**密集像素预测**的方法，输出的分辨率不会受到 **top-level** 采样的限制。在 Mask R-CNN 中，如果要得到更准确的 mask 特征，就必须增加 RoIPooler 的分辨率，这样变回成倍增加 head 的计算时间和 head 的网络深度；
- 4) 推理时间稳定：Mask R-CNN 的推理时间随着检测的 bbox 数量增多而增多，BlendMask 的推理速度更快且增加的时间可以忽略不计。

#### 3.1.2.2 行为克隆原理分析

本设计中的行为克隆部分继承观察学习的思想，从第三人的角度观看演示，

将演示视频片段转化为活动的抽象表示(活动特征)<sup>[14]</sup>。基于活动特征的不可变性,在不同的观察视角、对象属性、场景背景和机械手形态下,跟随演示中学习任务<sup>[11]</sup>。

本任务中所完成的行为克隆任务为复制人类移动布料的过程,解决思路为通过 BlendMask 分割模型,实时获取布料位置信息,包括中心点坐标和朝向。处理整个行为序列可以得到布料的空间变化信息,基于此信息计算布料的运动轨迹,进而将轨迹转换为机械臂的运动。

### 3.1.3 基于深度强化学习的布料码垛实现

机器人布料码垛技能获得框架可以分为机器人模块,感知与控制模块,技能模型、与技能学习。整体模型框图如图 3-5 所示。机器人模块具有执行器与传感器,是直接与布料码垛环境交互的模块。感知与控制模块是机器人本体与技能模型之间的中介层,主要用来从提取传感器原始信号中的特征状态变量。技能模型进行决策、规划和预测,产生运动指令经控制模块控制机器人本体运动。技能模型参数的获取通过技能学习实现。

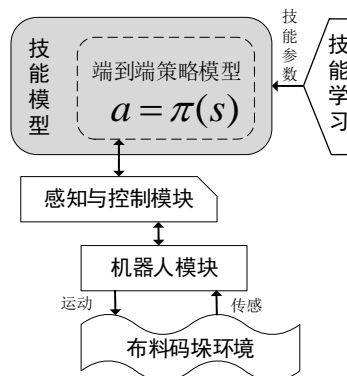


图 3-5 机器人布料码垛技能模型整体框图

机器人布料码垛实现的关键是技能模型构建与技能参数的学习。技能模型往往分为流程模型、运动模型以及策略模型。考虑到一方面当操作任务复杂时,操作流程往往较为繁琐,难以细致划分,另一方面机器人模块各运动变量存在非线性耦合关系,所以本文采用端到端的策略模型作为机器人技能获得模型<sup>[12]</sup>。

策略  $\pi$  是状态  $s$  到行动  $a$  的一个映射,即  $a = \pi(s)$ 。是机器人不断观察当前状态并做出反应行动,从而以闭环的形式执行操作技能。端到端策略将感知和控

制模块融合进策略模型中,可以直接将传感器原始信号端作为输入,并将底层机器人执行器的驱动信号端作为输出。解决了高维、复杂、耦合、非线性、连续状态空间下的技能获取。

把深度学习的感知能力与强化学习的决策能力相结合,利用深度强化学习可以使机器人技能获得更接近人类行为<sup>[13]</sup>。其基本学习过程包括高维度观察的获取与感知、动作的价值评价、策略空间-动作空间的映射、环境与动作交互并产生下一个动作等流程。图所示为深度强化学习算法框架模型示意图,其中,强化学习采用基于 Q 值函数离策略(off-policy)获取最优控制策略,深度学习采用多层深度卷积神经网络,通过深度全链接神经网络函数输出动作 Q 函数。深度强化学习算法框架包含深度卷积网络(策略网络)、深度卷积网络(估值网络)、完全链接神经网络(输出状态动作 Q 函数)等,输入历史数据并在 replay memory 中进行训练,以 Q 学习机制中的差值作为目标函数,从 replay memory 中进采样,进行 minibatch 的梯度求解。

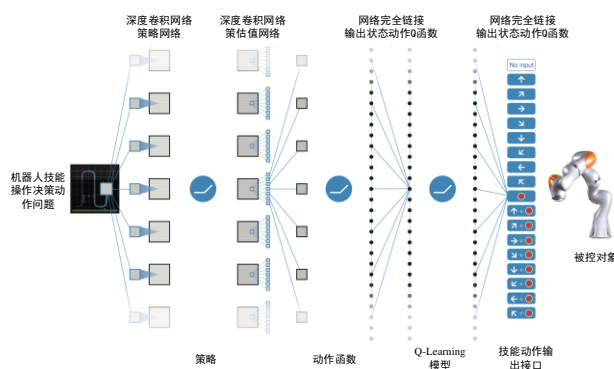


图 3-6 深度强化学习算法框架模型示意图

机器人布料码垛实现数学建模如下：

#### 1) Q学习建模：

设置初始所有 Q 值都等于 0, 利用探索(exploration)机制进行权衡, 采用 epsilon 贪婪策略, 设定某一随机策略为  $\pi$ ,  $Q(s,a)$  表征状态  $s$  下执行动作  $a$  时的 Q 值函数,  $\alpha$  表示学习率,  $R(s,a)$  表示表征状态  $s$  下执行动作  $a$  时获取的即时奖励,  $\lambda$  表示折现率,  $\max Q'(s,a')$  表示新状态下采取所有动作可能获取的最大未来 Q 值函数, 则 Q 学习更新公式如式(3.12)所示。

$$NewQ(s, a) = Q(s, a) + \alpha[R(s, a)] + \lambda \max_{a'} Q'(s, a') - Q(s, a) \quad (3.12)$$

基于式(3.12)，在每一个迭代周期可以更新 Q-table 中的一个值，单步进行，可以获得完整的 Q-table，则 Q 学习可以等价表示为：

$$Q^\pi(s, a) = E[R | s^t = s, a^t = a] \quad (3.13)$$

$$Q^\pi(s, a) = E_{s'}[r(s, a) + \gamma E_{a' \sim \pi}[Q^\pi(s', a')]] \quad (3.14)$$

基于式(3.13)和(3.14)可知， $Q(s, a)$  函数可以通过Q学习算法来估计，使用 Bellman 方程不间断迭代地更新  $Q(s, a)$  直至学习到完整策略。

## 2) DQN建模:

Q-table 中的每个数值表征在当前状态  $s$  下采用动作  $a$  可以获得的未来收益的折现和，通过不断迭代，可以最终使 Q-table 收敛，从而可以得出在任意状态下的最优策略。采用 Bellman 方程和 epsilon 贪婪策略，认为实际应该得到的奖励，目标是使 Q 值不断接近 Q-target。从 Q 学习的原理可知，其适用范围仅仅局限于状态和动作空间是离散且维数不高时，对于状态和动作空间是高维连续时，实现困难。引入 DQN 建模：

基于 Q 学习来确定 loss function 并使 Q-target 值和 Q-eval 值相差控制最小范围，DQN 中的损失函数是：

$$L_i(\theta_i) = E_{s, a \sim \rho(\cdot)} [(y_i - Q(s, a | \theta_i))^2] \quad (3.15)$$

计算如下：

$$y_i = E_{s' \sim \epsilon} [r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a] \quad (3.16)$$

由此，目标函数整体可以通过随机梯度下降方法进行优化：

$$\nabla_{\theta_i} L_i(\theta_i) = E_{s, a \sim \rho(\cdot); s' \sim \epsilon} [(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)) \nabla_{\theta_i} Q(s, a; \theta_i)] \quad (3.17)$$

## 3) 经验回放机制:

经验池的功能主要是解决相关性及非静态分布问题。具体做法是把每个时间步执行过程中 agent 与环境交互得到的转移样本  $(s_t, a_t, r_t, s_{t+1})$  储存到回放记忆单元，要训练时就随机拿出一些(minibatch)来训练，则有：

$$P(s' | s, a | \pi_1, \dots, \pi_N) \neq P(s' | s, a | \pi'_1, \dots, \pi'_N) (\pi_i \neq \pi'_i) \quad (3.18)$$

此时Q函数表示为在采用确定性策略 $\mu$ 下选择动作的奖励期望值，在机器人技能学习模型中采用DQN的结构使用Q网络来拟合Q函数，实现公式如下：

$$Q^{\pi}(s_t, a_t) = E[r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, \mu(s_{t+1}))] \quad (3.19)$$

## 3.2 硬件组成

本文设计的硬件部分由五部分组成：相机，云平台服务器，机械臂控制计算机、UR5e 协作机械臂、气动柔喙夹爪，系统各组件的作用以及数据流向如图 3-7 所示。



图 3-7 整体硬件组成

近端相机获取图像后输入到机械臂控制工控机，进行仿射变换后上传数据到云平台端进行实例分割。然后将机械臂需要抓取的实例目标下载到机械臂控制工控机，通过深度强化学习算法完成机械臂的抓取与投放。

远端相机获取操作者的动作，上传到云平台服务器，进行实例分割和行为克隆，向机械臂工控机传输抓取的目标的位置、朝向角度、大小等信息。

### 3.2.1 摄像头

本设计中的相机选用 Inter Realsense 435i 摄像头模组，用来感知视觉信息，如图 3-8 所示。该相机可以输出 1920\*1080 分辨率的彩色图像和分辨率为 1280\*720 的深度图像，并且配有 IMU 惯性模块，帧率可达 20 帧，采用

USB3.0 Type-C 接口与机械臂控制工控机进行连接。

本文所述系统中仅采用 RGB 信息，像素设置为 1920\*1080，主要用于感知机械臂工作平台上的视觉信息和人类的行为模式。

在本文中，远端相机与近端相机都由机械臂端的 Inter Realsense 435i 所实现。借由云平台的优势，远端相机可以部署在工作台区域以外的地方，以实现操作者在远端教授机械臂学习布料堆叠的过程。



图 3-8 Inter Realsense 435i

### 3.2.2 机械臂

本文采用如图 3-9 所示优傲公司生产的 UR 5e 系列机械臂。该机械臂功耗 250w，有效荷载 5 千克，工作半径 850mm，工有六个自由度，重复定位精度-/+0.03mm。本文通过 TCP/IP 协议利用 socket 进行数据通讯，利用 urx Python 功能包进行运动控制。

机械臂末端配置六个气动柔喙夹爪，由压缩空气驱动，输入气压 0.45~1Mpa，输出气压-70~120kPa，可通过遥控控制。额定电压 24VDC，额定功率 36W，采用 I/O，24VDC 电平信号控制。正压流量 140L/min，负压流量 25L/min。

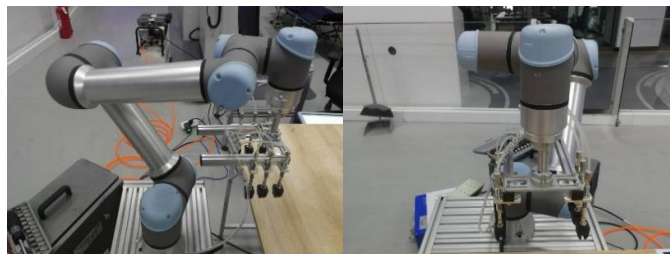


图 3-9 UR5e 机械臂

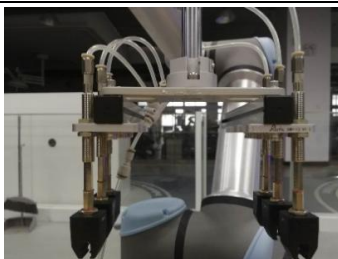


图 3-10 气动柔喙夹爪

### 3.2.3 机械臂端的工控机

机械臂端工控机操作系统为 64 位 Windows 10 专业版，处理器采用 Intel i7-9700k，频率 3.6GHz，运行内存(RAM)16GB，硬盘存储为固态 256G+1T 机械硬盘，显卡为 NVIDIA GeForce RTX 2070，如图 3-11 所示



图 3-11 机械臂端工控机

### 3.2.3 云服务器

本设计使用如图 3-12 所示的实验室搭载的服务器作为云服务器端，用于实例分割和行为克隆。实验室服务器搭载 i7 8700k CPU，三块 GTX1080Ti GPU，运行 ubuntu 服务器版系统。



图 3-12 云服务器

## 第四章 软件设计与流程

### 4.1 整体流程设计

本次设计主要实现两个功能：

- 1、相机自动识别随机堆叠在桌子上的布料，并自主完成布料的整理和堆叠。
- 2、通过对人类完成的动作进行行为克隆，操纵机械臂复现人类所完成的动作，并对动作进行存储，以实现机械臂堆叠的自动编程。

针对上述系统设计要求，整理设计程序框图如图 4-1 所示。

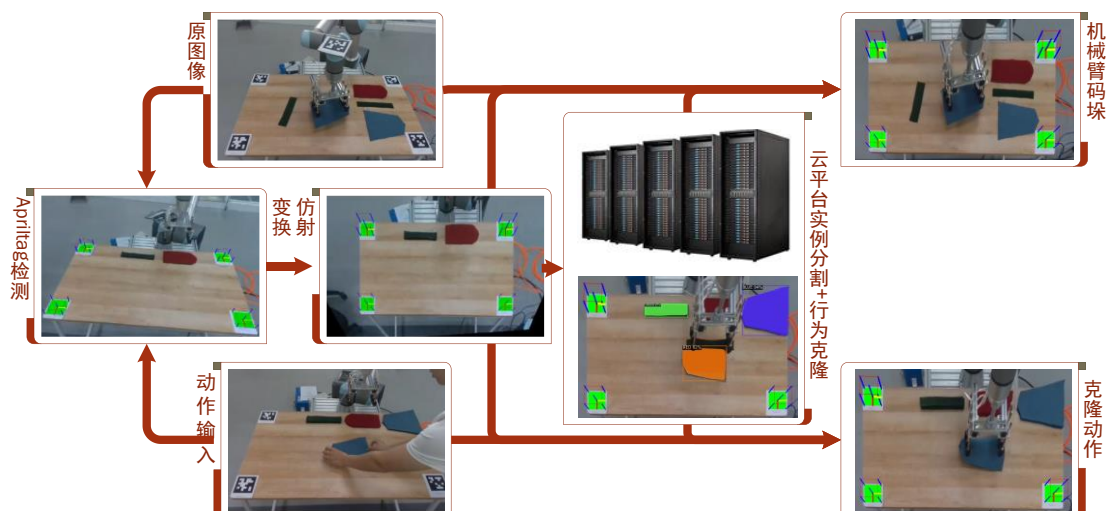


图 4-1 总体流程框图

算法在获取到原始图像后，首先在获取的图像中检测 Apriltag 标签，解算桌子四角在空间中的真实坐标点  $P_{1,2,3,4} = [X, Y, Z]$ 。如果其中一个标签被机械臂或人手所遮挡，通过其余三个标签的向量和即可对未成功识别标签的位置进行估算，此处假定 id 为 3 的标签被遮挡，左上、左下、右下、右上所贴标签的 id 分别为 0、1、2、3，列写公式为：

$$\vec{P}_{12} = P_2 - P_1 \quad (4.1)$$

$$\vec{P}_{10} = P_0 - P_1 \quad (4.2)$$



$$\vec{P}_{13} = \vec{P}_{12} + \vec{P}_{10} \quad (4.3)$$

$$P_3 = P_1 + \vec{P}_{13} \quad (4.4)$$

然后运用反投影公式,将桌子四角的三维点反投影到图像中获取桌子四角在图像中的像素坐标 $(u, v)$ 。

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4.5)$$

$f_x, f_y, c_x, c_y$  分别为相机在  $x, y$  轴上的焦距和光心,并使用桌子四角的原始像素坐标 $(u, v)$ 和到设定值坐标 $(u', v')$ ,解算仿射变换矩阵(本文设计中,设定坐标为俯视图下的桌子四角,每一个像素代表 1mm)。

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (4.6)$$

使用放射矩阵对图像中每个像素进行仿射变换即可将原始图像变换成俯视图视角。接下来,将解算成功的图像数据上传服务器,进行实例分割和人类的行为编码,求解机械臂的目标移动位置,并将目标数据下载到机械臂搭载的工控机。工控机使用仿射变换后的机载相机图像,通过强化学习完成布料的堆叠。

## 4.2 图像处理流程设计

图像处理是整个机械臂堆叠任务的核心,其处理的准确率和精度是整个抓取过程顺利进行的保证,此算法的流程图如图 4-2 所示。

首先,摄像头获取图像,使用仿射变换得到俯视视角下的布料分布图;然后,在俯视图的基础上,使用 BlendMask 算法,获取分割图;随后,最小包围框算法将计算每一个布料的中心点坐标 $(x, y)$ 和旋转角 $\theta$ ,送入行为识别模块,编码人类行为的动作执行过程,输出到机械臂完成动作复现,用已完成抓取任务。

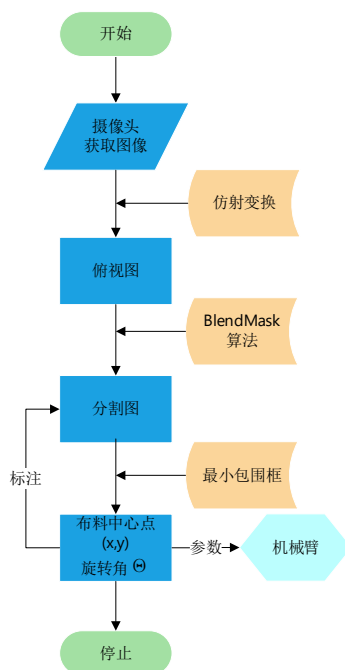


图 4-2 图像处理算法流程图

### 4.3 强化学习码垛流程设计

基于深度强化学习的机器人布料码垛模型学习流程包括环境图像数据预处理过程、网络训练过程、布料码垛测试等过程。流程路线如图 4-3 所示。

首先，数据预处理过程对机器人模块传感器采集到的图像数据与位置数据进行滤波处理，然后将处理后的布料位置信息与机器动作信息分类输入深度学习网络，进行特征选择。其次将选择后的特征一方面输入 DDON 网络的主网络中，作为训练数据，另一方面将特征作为一组经验存入经验池，作为 DDQN 网络的先验知识。随后进行网络训练，DDQN 网络共设置 1000 个 episode，每个 episode 过程通过利用特征数据进行损失函数梯度计算更新主网络参数，每迭代 200 个 episode 后将主网络的参数复制给目标网络，实现目标网络参数更新，完成所有网络训练步后将目标网络作为模型进行保存输出。最终，将机器人模块传感器采集到的图像数据与位置数据输入模型，模型输出布料码垛动作，完成机器人布料码垛任务。

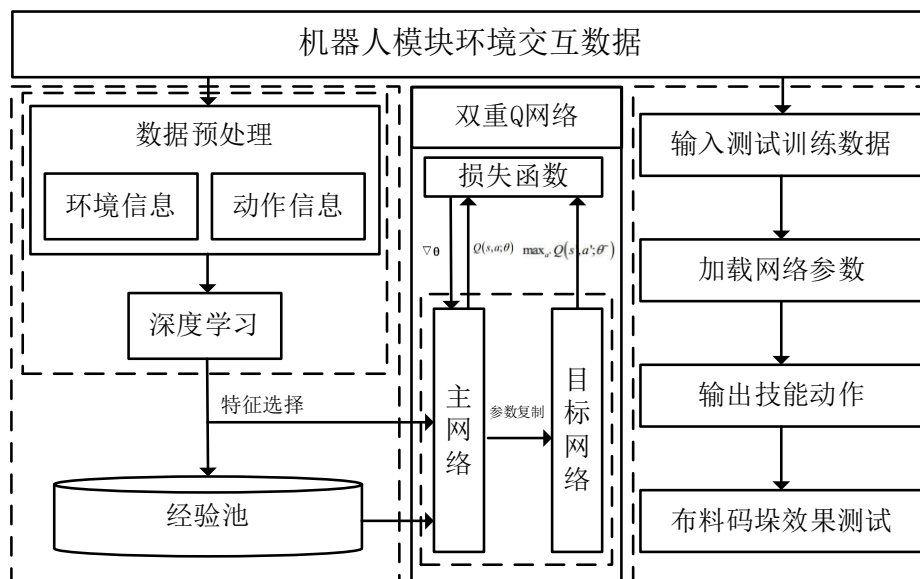


图 4-3 机器人布料码垛算法流程设计。

## 第五章系统测试与误差分析

### 5.1. 系统测试

#### 5.1.1 布料堆叠

相机获取原始图像后，检测贴在桌面四角的 Apriltag 标签，并使用标签的位置信息计算仿射矩阵，完成图像的仿射变换，如图 5-1 所示

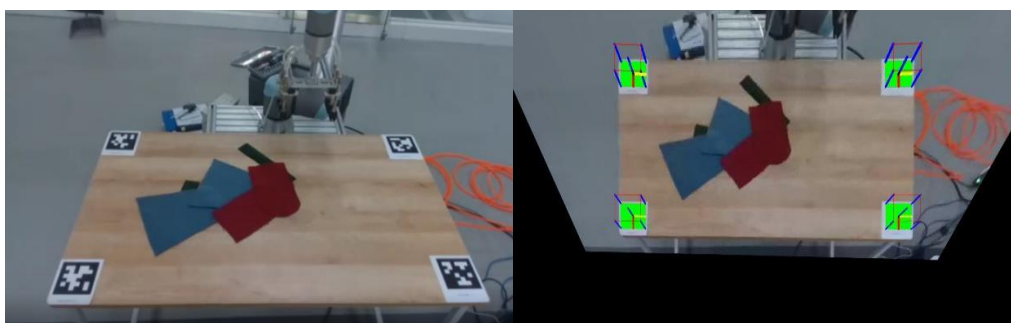


图 5-1 图像仿射变换

然后将仿射变换后的图像上传到云服务器，进行实例分割，获取图像的位置及朝向，如图 5-2 所示。

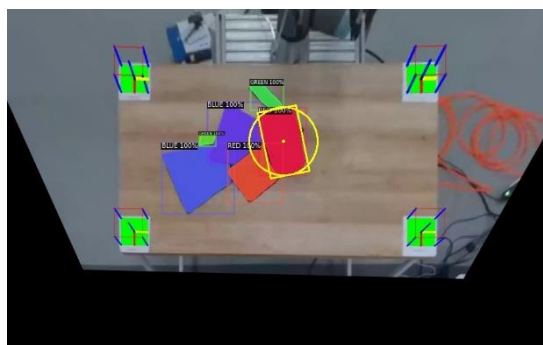


图 5-2 图像实例分割

并将实例信息返回到机械臂工控机完成实例信息的抓取，并将实例移动到设定的码垛位置，重复上述步骤完成布料的码垛，布料码垛过程如图 5-3 所示。

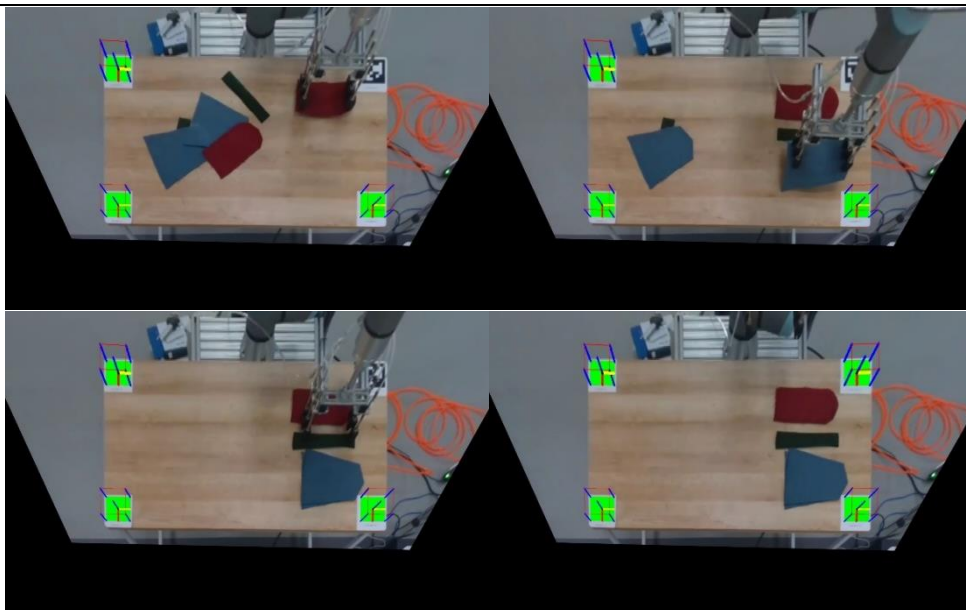


图 5-3 布料码垛过程

### 5.1.1 行为克隆

相机完成仿射变换后将图像上传云服务器，识别实例的空间位置，并完成人类行为的动作编码，输出人类动作在时间上的序列，如图 5-4 所示。

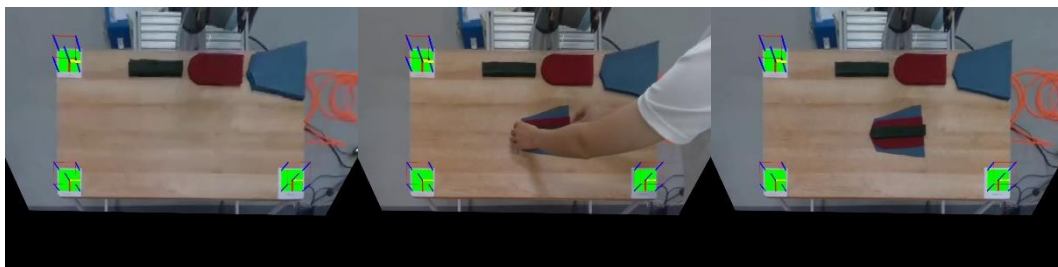


图 5-4 人类行为克隆编码

然后将该动作编码下载到机械臂工控机，按照时间解算动作编码，操作机械臂浮现人类动作，如图 5-5 所示。



图 5-5 机械臂复现行为克隆的编码

编码数据可视化如图所示，随着人手叠加布料，记录布料的移动数据，及叠加方式。图 5-6 左上、右上、左下、右下分别为视频中初始布料位姿数据、将第一块蓝色布料移动后的数据、将红色布料移动到蓝色布料上记录的数据、将绿色布料移动到红色布料上记录的数据。

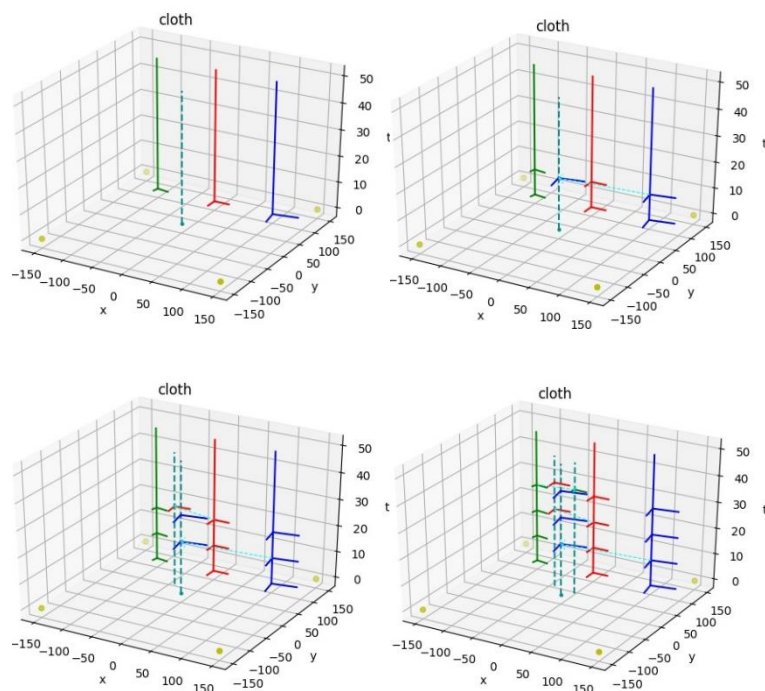


图 5-6 机械臂复现行为克隆的编码

## 5.2. 基于 AprilTag 码的仿射变换实验误差分析

仿射变化流程如图所示，首先通过相机拍摄到场景原图，然后识别定位场景中黏贴在桌子四个角上的 AprilTag 码，其中四个 AprilTag 码的标签分别为 0(左上)，1(左下)，2(右下)，3(右上)。最后结合 AprilTag 码，利用如图 5-7 所示的仿射变化实现图像校正。



图 5-7 仿射变化流程图

通过测量实际四个 AprilTag 码的距离，计算仿射变换后 AprilTag 码中心点的像素值坐标，结合每个像素值代表实际物理值距离 1mm 的变化关系，测试仿



射变化实验误差，结果如表 5-1 所示

表 5-1 仿射变化实验误差

标签 ID	标签像素坐标	理论计算距离	实际测量距离	误差
(0, 1)	(194,227), (192,625)	398mm	402mm	4mm
(0, 3)	(194,227), (993,225)	799mm	802mm	3mm
(1, 2)	(192,625), (996,630)	804mm	801mm	3mm
(2, 3)	(996,630), (993,225)	405mm	401mm	4mm

### 5.3 机械臂移动实验误差分析

机械臂移动误差通常包含有机械臂本体误差、机械臂手眼标定误差和测试真值测量误差。本设计首先对机械臂和相机进行手眼标定，在相机视场下拍摄 15 张不同姿态标定板照片，如图 5-8 所示，计算相机的内部参数：焦距及光心；相机到机械臂的外部参数：旋转矩阵和平移矩阵，如表 5-2 所示

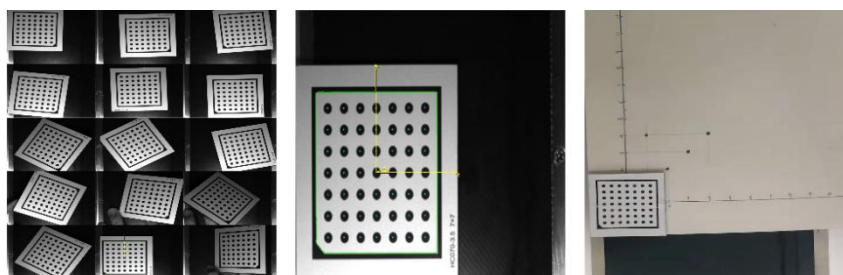


图 5-8 精度测试标定

跟据标定工具的误差输出，可以得到机械臂到相机的外参误差：

表 5-2 标定参数结果

参数	数据	单位
相机标定误差	0.0467	Pixel
R	(0.656, 359.961)	degree
T	(-22.521, -2.042)	mm

为确保机械臂布料码垛的准确性，本文设计实验指定工作台上左上、左中、左下、中上、中中、中下、右上、右中、右下 9 个点，使用 mm 精度尺子确定这 9 个点的真实位置，并将手眼标定后的机械臂移动到该指定点。通过如图 5-9 所示，机械臂的到达位置和该点的真实位置的误差，分析机械臂的移动误差。

在本设计中，布料厚度可以忽略，研究工作主要针对工作台平面展开，机械臂运动环境被限制于一个水平面。因此，在此部分仅考虑工作台平面  $x, y$  轴上的机械臂运动误差，机械臂运动误差如表 5-3 所示。

表 5-3 机械臂运动误差

理论计算位姿	实际检测位姿	X 轴位置偏差	Y 轴位置偏差
(152.458, 245.587)	(152.953, 245.356)	0.495mm	0.231mm
(171.821, 143.235)	(171.411, 142.976)	0.410mm	0.259mm
(131.254, 342.871)	(131.695, 343.122)	0.441mm	0.251mm



图 5-9 到达位姿测试

机械臂码垛最终状态如图 5-10 所示，底层布料编号为 1，上层布料编号为 2，先后测量三种布料上下两层单布料的中心点与旋转角度，机械臂布料码垛误差如表 5-4 所示。



图 5-10 机械臂码垛最终状态。

表 5-4 机械臂布料码垛误差

布料 ID	中心点像素坐标	旋转角度	中心点误差	旋转角度误差
(red1,red2)	(903,325) (904,321)	-0.531, -0.545	4.12mm	0.014rad
(green1,green2)	(945,467) (948,464)	0.437, -0.521	4.24mm	0.958rad
(bule1,bule2)	(976,527) (970,523)	-0.654, -0.589	5.65mm	0.650rad

由表 5-3 与表 5-4 可得机械臂理论计算值与实际检测值误差小于 0.5mm，布



料堆叠小于 1mm，旋转角度小于 1rad，满足作业要求。误差存在原因：一方面是相机在畸变矫正过程中存在微小的矫正误差，另一方面位置到达探针在安装过程中，本身具有非垂直偏差。

## 5.4 强化学习布料堆垛学习误差分析

在本设计当中的布料堆叠工作中，深度强化学习网络结构采用具有 100 层隐藏层的全连接网络。经验池大小为 2000，训练 batch 为 64，学习率为 1，折扣因子为 0.95，模型更新频率为 200，共 1000 episode。

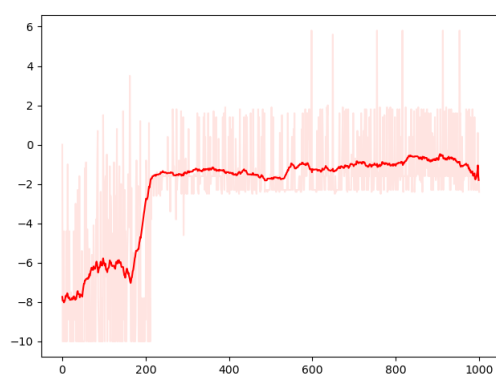


图 5-11 机器臂布料码垛奖励值变化曲线

由图 5-11 训练结果可以看出，网络在第 200 个 episode 后才真正开始训练。机械臂通过 200 个 episode 探索收集环境后，累计动作奖励值提高 7 分，到第 800 个 episode 时，累计奖励值趋于稳定收敛，模型获取较好的训练效果。训练过程如图所示：

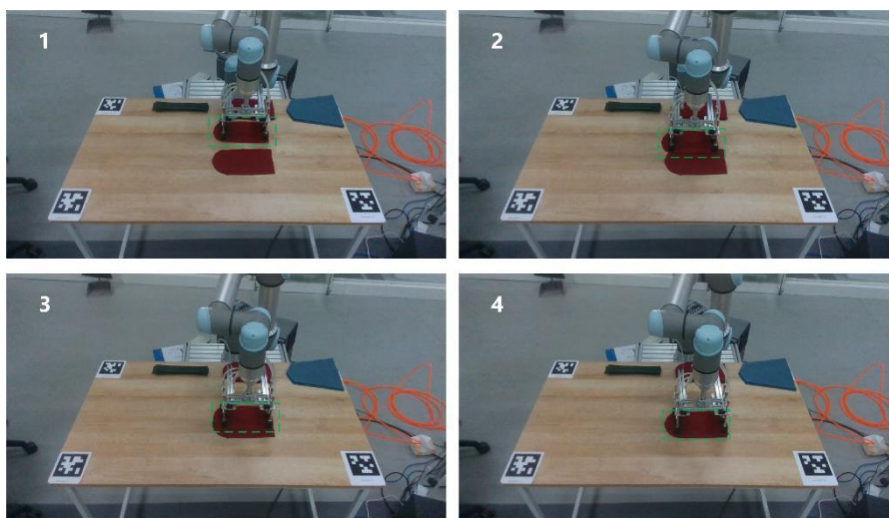


图 5-12 机器臂布料码垛学习过程图

针对前 200 个 episode 奖励函数较低的情况，可能是由于前期机械臂采用随机动作积攒经验值，动作奖励值处于较低水平。针对该现象可以考虑后期加入人工示教的先验知识，提高前期环境探索的有效性。

在整体的训练过程中，平均奖励值曲线处于负数。原因可能是奖励函数设置离散，针对该现象后期可以考虑通过分段奖励函数或者符号奖励机制。

## 5.5 基于 BlendMask 的观察学习误差分析

### 5.5.1 图像标注误差

训练 BlendMask 模型所使用的数据集均为手工标定，所用工具为 VIA 标注助手，因此在数据集中存在标注错误的样例，会对模型的训练过程产生负面影响。

### 5.5.2 模型误差

送入 BlendMask 分割算法的图像是经过放射变换的摄像头拍摄的图像，因此整个模型的误差受放射变换误差的影响，且 BlendMask 模型本身存在检测误差。

在运行 BlendMask 模型过程中，batchsize 大小设为 10，学习率为 0.0025，最大迭代步数为 600 步，在训练过程中所产生的 loss 损失变换图如图 5-13 所示。



图 5-13 训练 loss 损失变换图

该损失值包含类别损失、包围框损失、像素损失等，由图示可以看出，通过逐次的迭代，训练的损失逐步下降趋于稳定，算法可以完成图像分割工作。

## 总结

本次设计围绕纺织工业中布料堆叠工作中存在的三个问题展开：

- 1) 当前布料堆叠工作当中，每次更换布料堆叠方案，都需要人工重新编程机械臂动作；
- 2) 当前主流布料堆叠方法基于固定位置编码，对布料摆放位置的精度要求较高。
- 3) 当前纺织工业人机协同工作中，需要相机处于不同位置以获取更大的视野。而如果需要相机可以进行移动，则相机如何在空间中进行固定也是很棘手的问题，且相机的晃动震动等问题都会对布料位置的识别产生影响。

针对这些问题，本文设计了人机协同环境下基于行为克隆的云基机械臂布料堆叠系统，通过工作台黏贴 **Apriltag** 码的方式将工作台区域仿射变换到图像中固定区域，以应对相机的运动旋转带来的视角变化。之后，本文在仿射变换后的图像中进行实例分割，检测布料在工作台上的位置。通过在实例分割获取的实例上求取最小包围框，计算实例的中心点、尺寸和位姿朝向。最后通过深度强化学习方法完成布料的堆叠。并将上述算法部署到云服务器上，以满足算法对高性能 GPU 的需求。

对于每次更换布料堆叠方式都要重新编程机械臂动作流的问题，本文设计实现了行为克隆算法，在时间流上编码人动作对工作台区域布料位姿变化产生的影响，然后通过机械臂端对该动作进行解码，完成行为克隆的任务。

但是，在本次设计当中依旧存在一些问题，受限 **Apriltag**、机械臂、相机畸变等带来的误差，本文的布料堆叠工作当中依旧存在一些误差，且在视野中只能检测到三个 **Apriltag** 时，对相机获取图像的仿射变换也会存在一些误差。在未来进一步的研究当中，计划对系统进行进一步的改进，建立 **Apriltag**、机械臂的运动误差模型，联立相机的畸变矫正误差，进一步缩小系统中存在的误差。

其次，本设计中的行为克隆方法采用了一种 **two-stage** 的方案，在图像处理中先进行实例分割，然后再进行行为编码，会耗费较高的运算时间，产生一定程度的编码误差。在下一步的设计中，计划采用端到端的方式模仿学习人类的行为模式。

## 参考文献

- [1] Otsu N. A Threshold Selection Method from Gray-Level Histograms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66.
- [2] Ciresan D, Giusti A, Gambardella L M, et al. Deep neural networks segment neuronal membranes in electron microscopy images[C]//Advances in neural information processing systems. 2012: 2843-2851.
- [3] D. Beier, D. Gulamhussene, C. Bode, G. Gulamhussene, and C. Steinmann, "Method For Automatically Producing An Optical Blend Mask," US Patent US 2019/0342530 A1, 2019/11/07, 2019. Available: <https://lens.org/020-402-431-942-110>.
- [4] Hata S, Hayashi J, Hojoh H, et al. Design of cloth handling robot system[C]. international conference on industrial technology, 2010: 1611-1616.
- [5] Perez L A, Rodriguez I, Rodriguez N, et al. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review.[J]. Sensors, 2016, 16(3).
- [6] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. arXiv: Learning, 2013.
- [7] AprilTag:A robust and flexible visual fiducial system. Edwin Olson. Robotics and Automation(ICRA),IEEE International Conference on . 2005
- [8] Canny J. A Computational Approach to Edge Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8(6): 679-698.
- [9] Chen H, Sun K, Tian Z, et al. BlendMask: Top-down meets bottom-up for instance segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 8573-8581
- [10] 胡立华,张继福.基于图像的四边形自动检测方法[J].小型微型计算机系统,2014,35(08):1911-1915.
- [11] Pauly L, Agboh W C, Hogg D C, et al. One-Shot Observation Learning Using Visual Activity Features[J]. arXiv preprint arXiv:1810.07483, 2018.
- [12] Mason K, Grijalva S. A review of reinforcement learning for autonomous building energy management[J]. Computers & Electrical Engineering, 2019: 300-312.

- [13] K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," in IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26-38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.
- [14] Bicho E, Louro L H, Erhagen W, et al. Integrating verbal and nonverbal communication in a dynamic neural field architecture for human–robot interaction[J]. Frontiers in Neurorobotics, 2010.
- [15] Graham R L, Yao F F. Finding the convex hull of a simple polygon[J]. Journal of Algorithms, 1983, 4(4): 324-331.
- [16] Suzuki S, Abe K. Topological Structural Analysis of Digitized Binary Images by Border Following[J]. Graphical Models & Image Processing, 1985, 30(1): 32-46.
- [17] Garcíagarcía A, Ortsescolano S, Oprea S, et al. A Review on Deep Learning Techniques Applied to Semantic Segmentation[J]. arXiv: Computer Vision and Pattern Recognition, 2017.
- [18] Lu H, Li Y, Yang S, et al. Fast Active Contours Model for Image Segmentation: A Review of Recent Trends[J]. Journal of the Institute of Industrial Applications Engineers, 2013, 1(1): 15-21.
- [19] Ji, W., Wang, L. Industrial robotic machining: a review. Int J Adv Manuf Technol 103, 1239–1255 (2019). <https://doi.org/10.1007/s00170-019-03403-z>
- [20] E. Olson, "AprilTag: A robust and flexible visual fiducial system," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 3400-3407, doi: 10.1109/ICRA.2011.5979561.