

# 基于微流控芯片和机器视觉的药物筛选研究<sup>1</sup>

## 摘要

目前，环境中化学品的数量日益增多，对人们的身体健康造成严重威胁，如何科学的对这些化学品进行风险评估是公共健康领域的热点与难点。传统的动物实验虽然研究结果真实可靠，但具有周期长、成本高以及涉及动物保护等局限性，难以进行高通量和大规模的毒性研究；使用细胞或者类器官等虽可进行较高通量的毒性评价与筛选，但其研究结果无法代替动物体内实验。秀丽隐杆线虫具有体积小、生命周期短、结构简单和高基因保守性等特点，是环境暴露和毒理学研究领域中重要的体内研究工具。微流控体系与线虫大小尺度相匹配，与传统的在琼脂板上进行的线虫实验相比，其具有反应体系小、通量高、自动化且操作灵活等优势。因此基于微流控芯片的线虫研究平台为高通量、大规模的环境化学品评估提供了新的手段。然而目前仍缺乏集成自动化操控及图像分析的一体化微流控线虫分析平台。

针对上述问题，本文首先搭建了一个基于微流控芯片的硬件平台，可以在芯片上进行线虫的简单培养以及为线虫在给定环境毒素下的暴露创造一个精确的微环境。在自动化图像分析线虫的毒理实验中，体长和摆动频率往往作为评价毒性的重要生理指标，然而图像分析存在下述难点：1. 由于线虫通体透明且由 PDMS 制作的微流控芯片也是透明的，因此线虫的前景轮廓分割是整个系统中的难点。2. 当一个腔室中存在多个线虫时，多个线虫轮廓可能会纠缠在一起，由于无法辨识单个线虫的轮廓，从而导致跟踪丢失。3. 由于线虫是非刚体其形态变化多种多样，线虫在多帧图像中的跟踪也是一个难点。针对上述问题，本论文提出如下解决方案：1. 针对传统的图像分割算法在线虫前景轮廓分割任务中存在的不足（如：面临鲁棒性差、依赖超参数选择和分割效果不理想等问题）。本文提出了一种基于深度卷积网络和条件随机场的分割算法。通过与其他分割算法对比，表明本文提出的算法能够显著地改善线虫前景轮廓分割的性能。在我们标注的线虫前景分割数据集上，本文提出的分割算法在像素误差指标上达到 0.11% 的分割精度。2. 为了解决多线虫轮廓之间的纠缠，本文提出了一种基于卷积神经网络的编解码器架构，

---

<sup>1</sup>本研究由上海市科委基础重大研发计划(17JC1401001)资助

该模型有效的解决了单个线虫轮廓的解析问题。3. 针对线虫的跟踪问题，本文提出了一种简单有效的跟踪策略，通过最近邻搜索匹配的方式实现线虫的跟踪。实验发现，在线虫轮廓分割较为完整的情况下，该跟踪方法能够对线虫轮廓实现鲁棒地跟踪。利用跟踪的线虫轮廓，可以计算出线虫的体长、摆动频率等生理特征。最后，通过该自动化平台研究了线性浓度梯度双氧水对线虫活性地影响，实验结果和传统的实验结果一致，展示了本文提出的基于微流控平台和自动化图像分析系统在毒理学研究中的应用前景。

**关键词：**秀丽隐杆线虫；微流控；药物筛选；轮廓分割；卷积网络

# DRUG SCREENING BASED ON MICROFLUIDIC CHIP AND MACHINE VISION

## ABSTRACT

Currently, the number of chemicals in the environment is increasing and posing a serious threat to people's health. How to conduct risk assessment of those chemicals scientifically is a hotspot and a difficult point in the public health field. Although traditional animal experiments are reliable and have great significance, they have long cycle times, high costs, and limitations related to animal protection, making it difficult to conduct high-throughput and large-scale toxicity studies. Cells or organs can be used in high throughput toxicity evaluation and screening, but the results of the study cannot replace experiment *in vivo*. The representative model of *Caenorhabditis elegans*(*C.elegans*) has the characteristics of small size, short life cycle, simple structure and high gene conservation. It is an important *in-vivo* research tool in the field of environmental exposure and toxicology research. The microfluidic system is matched with the size of the *C.elegans*. Compared with the *C.elegans* experiment on the agar plate, it has the advantages of small reaction system, high throughput, automation and flexible operation. Therefore, the microfluidic chip-based *C.elegans* research platform provides a new means for high-throughput, large-scale environmental chemical assessment. However, there is still no integrated microfluidic analysis platform that integrates automated opration and image analysis.

In response to the above problems, this paper first builds a hardware platform based on microfluidic chip, which can realize the simple cultivation of *C.elegans* and create a precise microenvironment for the exposure of *C.elegans* to a given environmental toxin. In the automated image analysis of toxicology experiments with *C.elegans*, body length and swing frequency are often used as important physiological indicators for evaluating toxicity. However, image analysis has the following difficulties: *a)* Since the *C.elegans* is transparent and the microfluidic chip made by PDMS is also transparent, the foreground contour segmentation of the *C.elegans* is a difficult point in the whole system. *b)* When multiple *C.elegans* are present in one chamber, multiple contours of *C.elegans* may be

entangled, resulting in loss of tracking due to the inability to recognize the contours of individual C.elegans. *c)* Because C.elegans are non-rigid and their morphological changes are diverse, it is also a difficult point to track the C.elegans in multi-frame images. To alleviate the problems mentioned above, this paper proposes the following solutions: *a)* Aiming at the shortcomings of the traditional image segmentation algorithm in the foreground segmentation task of the worm (such as: poor robustness, dependence on hyperparameter selection and unsatisfactory segmentation), this paper proposes a segmentation algorithm based on deep convolutional networks and conditional random fields. Compared with other segmentation algorithms, the proposed algorithm can significantly improve the performance of foreground segmentation of C.elegans. On the segmentation dataset of the C.elegans we have labeled, the segmentation algorithm proposed in this paper achieves a segmentation accuracy of 0.11% on the pixel error index. *b)* In order to solve the entanglement between the contours of multi-worms, this paper proposes an encode-decode architecture based on convolutional neural network, which effectively solves the problem of parsing the contour of a single C.elegans. *c)* In order to achieve multi-worms tracking, this paper proposes a simple and effective tracking strategy to track the C.elegans by nearest neighbor search. The experiment found that the tracking method can achieve robust tracking of the C.elegans under the condition that the foreground segmentation of the C.elegans is complete. Using the traced contour of C.elegans, physiological characteristics such as body length and swing frequency of the C.elegans can be calculated. Finally, by studying the effect of linear concentration gradient hydrogen peroxide on C.elegans activity and the results are consistent with the traditional experiment results. The application prospects of microfluidic platform and automated image analysis system in toxicology research are presented.

**KEY WORDS:** C. elegans, Microfluidics, Drug screening, Contour segmentation, Convolutional network

## 目 录

<b>第一章 绪论</b>	<b>1</b>
1.1 论文研究的背景及意义 . . . . .	1
1.2 秀丽隐杆线虫概述 . . . . .	1
1.3 国内外研究现状 . . . . .	3
1.3.1 微流控芯片线虫操控方法 . . . . .	4
1.3.2 药物筛选与毒理实验 . . . . .	8
1.3.3 机器视觉在线虫研究中的应用 . . . . .	9
1.4 存在的问题 . . . . .	10
1.5 研究内容 . . . . .	11
1.6 论文章节安排 . . . . .	11
<b>第二章 微流控线虫芯片的设计及硬件平台的搭建</b>	<b>13</b>
2.1 引言 . . . . .	13
2.2 双层线虫梯度芯片的设计及制作 . . . . .	13
2.2.1 线虫梯度芯片的设计 . . . . .	13
2.2.2 实验材料与仪器 . . . . .	14
2.2.3 芯片模具加工工艺 . . . . .	14
2.2.4 线虫梯度芯片的制作 . . . . .	15
2.2.5 振荡的理论机制 . . . . .	17
2.2.6 染料与荧光实验 . . . . .	17
2.3 单层侧向阀门线虫培养芯片的设计及制作 . . . . .	19
2.3.1 单层阀门线虫培养芯片的设计 . . . . .	19
2.3.2 单层阀门芯片模具的制作 . . . . .	19
2.3.3 单层阀门芯片的制作 . . . . .	20
2.3.4 单层侧向阀门线虫固定芯片的测试 . . . . .	20
2.4 阀门控制系统搭建 . . . . .	20
2.5 高速视频采集程序的设计 . . . . .	21
2.6 本章小结 . . . . .	23

---

<b>第三章 自动化线虫视频分析与特征提取</b>	<b>25</b>
3.1 引言 . . . . .	25
3.2 线虫图像处理总体方案介绍 . . . . .	25
3.3 卷积神经网络介绍 . . . . .	26
3.3.1 卷积层 . . . . .	27
3.3.2 池化层 . . . . .	28
3.3.3 Dropout 层 . . . . .	28
3.3.4 Batch Normalization 层 . . . . .	28
3.4 基于深度卷积网络的线虫前景轮廓分割 . . . . .	29
3.4.1 传统的图像分割方法分类介绍 . . . . .	30
3.4.2 数据集的制作 . . . . .	32
3.4.3 条件随机场模型在分割任务中的应用 . . . . .	33
3.4.4 网络结构的设计 . . . . .	35
3.4.5 网络模型的训练 . . . . .	35
3.4.6 实验结果分析 . . . . .	38
3.5 基于深度卷积网络的线虫轮廓解析 . . . . .	41
3.5.1 线虫轮廓解析方法介绍 . . . . .	41
3.5.2 随机点的产生 . . . . .	43
3.5.3 SingleOut-Net 网络输出后处理 . . . . .	45
3.5.4 数据集的制作 . . . . .	46
3.5.5 网络结构的设计 . . . . .	46
3.5.6 评价指标 . . . . .	47
3.5.7 网络的训练 . . . . .	48
3.5.8 测试结果与分析 . . . . .	49
3.6 线虫轮廓的跟踪 . . . . .	50
3.7 线虫的特征提取 . . . . .	52
3.7.1 线虫轮廓中间脊线提取 . . . . .	52
3.7.2 身体弯曲角度的计算以及摆动频率的估计 . . . . .	53
3.8 本章小结 . . . . .	54
<b>第四章 线虫的特征提取和急性氧化应激实验</b>	<b>55</b>
4.1 引言 . . . . .	55
4.2 线虫的氧化急性应激实验 . . . . .	55
4.2.1 线虫的同步化 . . . . .	55

4.2.2 线性梯度稀释芯片的操作 . . . . .	55
4.2.3 秀丽隐杆线虫急性氧化应激实验结果 . . . . .	56
4.3 本章小结 . . . . .	56
<b>第五章 总结与展望</b>	<b>58</b>

## 第一章 绪论

### 1.1 论文研究的背景及意义

20世纪以来，人类生存环境中化学品日益增多，针对化学物进行健康风险的评估是公共健康领域当前重要的研究课题。目前人类与环境化合物的接触方式多表现为低剂量下的复合暴露，数量巨大且种类复杂，如何实现高通量的毒性评价和科学的风险评估成为毒理研究的热点与难点。

自2005年起，美国启动了21世纪毒理学研究计划<sup>1</sup>（TOX21），主要借助体外测试和计算生物信息学等方法进行高通量化学品毒性评估。然而，使用细胞或类器官的体外方法虽然可以进行较高通量的毒性测试，但无法代替动物体内实验，因为其结果无法反映化合物进入生物体内吸收、分布、代谢与排泄环节对毒性的影响。传统的动物体内毒性研究结果具有很大的参考意义，但是实验周期长、成本高，并且涉及动物保护与伦理。如果进行待测化合物低剂量复合暴露毒性研究，需要不同化合物的多剂量进行配伍，基于上述动物实验的局限性，难以进行高通量的毒性测试。秀丽隐杆线虫作为一种模式生物，对环境变化非常敏感；其生命周期和寿命较短，身体尺寸较小易操作，以大肠杆菌为食易培养，身体透明易观察，染色体和基因组较小易分析，生物学信号通路保守，因此近年来线虫被广泛应用于重金属、PM2.5、农药、纳米颗粒和有机污染物等环境毒理学研究领域<sup>[1-5]</sup>。国际上目前已进展到第三阶段的TOX21计划也在使用包括线虫在内的模式生物对前期结果进行验证，并计划对更多环境化合物进行毒性筛选。

传统的线虫培养方法在液体或琼脂板上进行，需要的线虫数量多，待测化合物所需量较大，此外难以对单个线虫实现精确刺激、操控和追踪。微流控体系与线虫大小尺度相匹配，有望实现线虫芯片内培养、成像和数据分析等。自从2011年美国启动“类器官芯片”计划之后，以微流控芯片为平台的毒理测试技术发展迅速，相比器官芯片平台，线虫芯片是在动物个体水平上进行毒理学研究，具有不可替代的优势。

### 1.2 秀丽隐杆线虫概述

秀丽隐杆线虫（简称 *C. elegans*）是一种微米尺度的无脊椎模式生物，主要生活在土壤中或水中，以大肠杆菌 OP50 为食，成虫体长约为 1mm 宽度 50um，并且从

---

<sup>1</sup><http://tox21.org/>

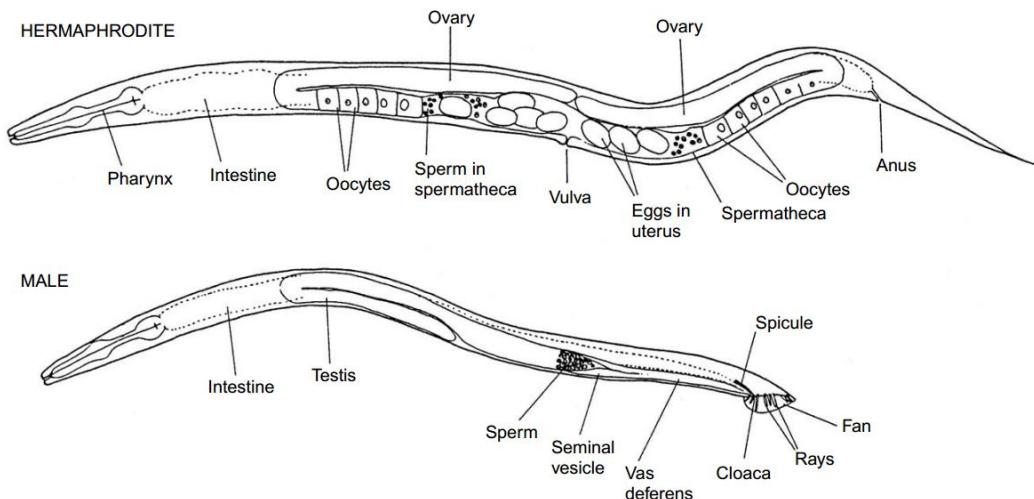


图 1-1 秀丽隐杆线虫解剖图

Figure 1-1 Anatomy of adult *C.elegans*.

幼虫长到成虫只需要 3 天。秀丽隐杆线虫是线虫动物门中的一种，包含雌雄同体和雄性两种性别如图1-1。两种性别都是二倍体，都有五对常染色体，但雌雄同体的线虫有两个 X 常染色体 (XX) 而雄性线虫只有一个 X 常染色体 (XO)。在自然条件下，线虫几乎都为雌雄同体，雄性个体只占约五百分之一。线虫的生命周期较为简单如图1-2所示，成年雌雄同体线虫同时含有精子和卵母细胞，因此能够自体受精。每一个自体受精的雌雄同体线虫大约可以产生 330 多个蛋。如果和雄性个体交配产蛋量将会提高，并且有利于基因的重新组合。在 25°C 下孵化 12 个小时即可得到 L1 期的线虫幼虫，长度约为 0.15mm。

L1 期幼虫基本上具备了和成虫一样的器官结构，除了还没有形成生殖器结构，从 L1 期线虫长到 L4 期只需要 3 天。线虫作为一种现代动物模型被广泛地用于细胞生物学、神经科学、衰老与发育和毒理学等研究中。在 2002 年，研究秀丽线虫的研究团体就已经扩展到 300 多个实验室，分布在 20 多个国家和地区。1998 年，构成整个基因组的 9700 万个碱基对 DNA 的测序基本完成，这是第一个经过完整基因组测序的多细胞生物。与其他模式生物相比秀丽隐杆线虫具有如下技术优势：

- a) 由于线虫通体透明，可以通过荧光蛋白标记的表达观察细胞的分裂等许多重要的生理过程，还可以用于对神经元成像，研究外部刺激与神经元活动之间的关联。
- b) 线虫的生命周期短、培养简单以及繁殖能力强，这些特性都有助于加快实验周期，提高实验的并行性。
- c) 线虫身体构造简单，遗传模式保守，有利于减小实验中由于个体差异而引

起实验结果的扰动。

- d) 对线虫基因的测序已经完成，使得线虫可以在基因筛查分析和其他遗传学实验中发挥重要作用。
- e) 线虫基因与人类的基因有约 40% 的同源性，许多的研究者将线虫作为重要疾病模型研究疾病机制。

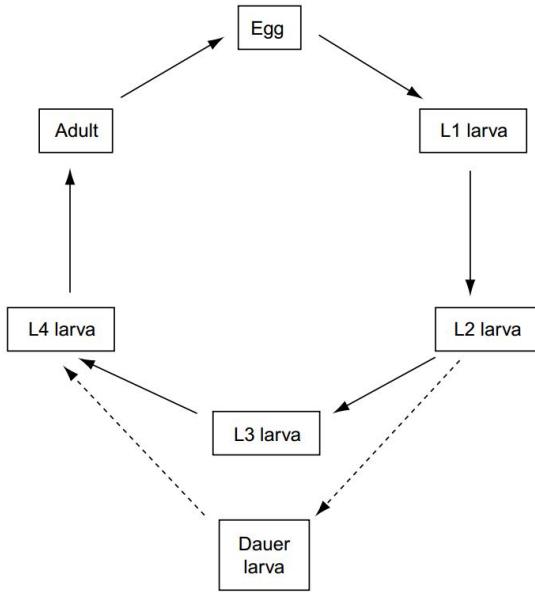


图 1-2 秀丽隐杆线虫生命周期

Figure 1-2 The life cycle of *C.elegans*.

### 1.3 国内外研究现状

微流控是一种在微米尺度操纵流体的技术，具有反应体系小、通量高、自动化且操作灵活等优势，被越来越多的应用于细胞和微米尺度生物的研究中。微流控技术在线虫研究中的应用为研究者们提供了一个全新的研究平台，极大的促进了相关领域的研究进展。目前国外课题组主要是乔治亚理工大学的 Lu hang 教授课题组和洛桑联邦理工学院 Gijs Martinus 教授的微系统实验室，国内主要是中科院大连物化所的秦建华课题组和华中科技大学的刘笔锋教授课题组。利用微流控技术研究线虫具有的优势有：1. 利用微流控芯片可以实现在细胞尺度上对线虫的操纵。2. 利用微流控芯片可以实现对线虫的快速固定与成像，与使用药物麻醉的方法相比，这种固定方式不会对线虫产生任何损害，线虫可以在后续步骤中恢复。3. 利用微流控芯片可以快速的从上千只线虫中的筛选出需要的表型，实现线虫的快速分选。4. 微流控芯片可以为线虫的培养提供精确的微环境，为线虫的感知实验

提供精确的刺激传达。本文将从以下几个方面对近年来国内外相关领域的发展状况进行回顾，包括微流控线虫操纵方法、线虫药物筛选以及介绍机器视觉方法在线虫研究中的应用。

### 1.3.1 微流控芯片线虫操控方法

在线虫相关的实验中，经常涉及一些对线虫的日常操纵。如线虫的片上培养、线虫分选、线虫的固定与成像等。为了避免复杂的人工操作，研究者们提出了许多不同用途的芯片，这些芯片能够将不同的操作集成在一起，极大的提高了实验的并行性和自动化程度。

#### 1.3.1.1 线虫的培养

线虫的长期培养是研究线虫的衰老和发育的基础，在许多需要对线虫进行长期观察的实验中也需要解决线虫的长期培养问题。线虫的长期培养需要与外部环境进行物质交换，包括氧气和食物的供应以及将代谢废物排出。微流控芯片的制作材料一般为聚二甲基硅氧烷 (polydimethylsiloxane, PDMS)，这种材料具有很好的透气性，因此微流控芯片可以为线虫提供氧气。其中一个早期的工作是由 Kim 等人<sup>[6]</sup> 完成的，如图1-3(a) 所示，他们设计了一款类似于 CD 形状的微流控芯片可以实现线虫的片上培养长达数天。食物会在离心力的作用下进入到线虫的培养腔室。线虫腔室中的废物会在离心力的作用下被甩出，从而实现了物质的交换。这款芯片可以将线虫培养到三代并且不会影响到线虫的生长和行为。然而，由于其芯片结构设计较为简单，所以无法区分不同代之间的线虫且难以实现对单个线虫的跟踪和成像。

针对以上的局限，Hulme 等人<sup>[7]</sup> 设计了一款新的线虫培养芯片如图1-3(b) 所示。由一排平行的腔室组成，可以实现线虫的长期培养及研究其运动行为。利用侧边的楔形通道来固定线虫可以实现线虫的固定与成像以及监测其体长的变化。可以同时培养 16 条成虫，极大的加快了衰老相关的研究，并发现线虫的摆动频率会随着线虫发育日渐成熟而出现下降。另一方面，研究者们将液滴和可逆凝胶运用在线虫的长期培养中<sup>[8-11]</sup>。Krajniak 和 Lu<sup>[9]</sup> 开发了一款集成微流控芯片如图1-3(c) 所示，该芯片由 8 个微腔室构成，通过周围的微管道和阀门控制。成功地展示了在一块芯片上进行线虫地培养、固定与成像等操作。线虫由一个入口通道进入到 8 个线虫培养腔室，并将 PF127 可逆凝胶经入口通道注入到线虫培养腔室。这种聚合物在低温 (大约 15°C) 下呈现较高的黏度，在高温下 (大约 21°C) 呈现胶状。运用这款芯片，成功实现了从 L1 期线虫到成虫过程的生理监测。

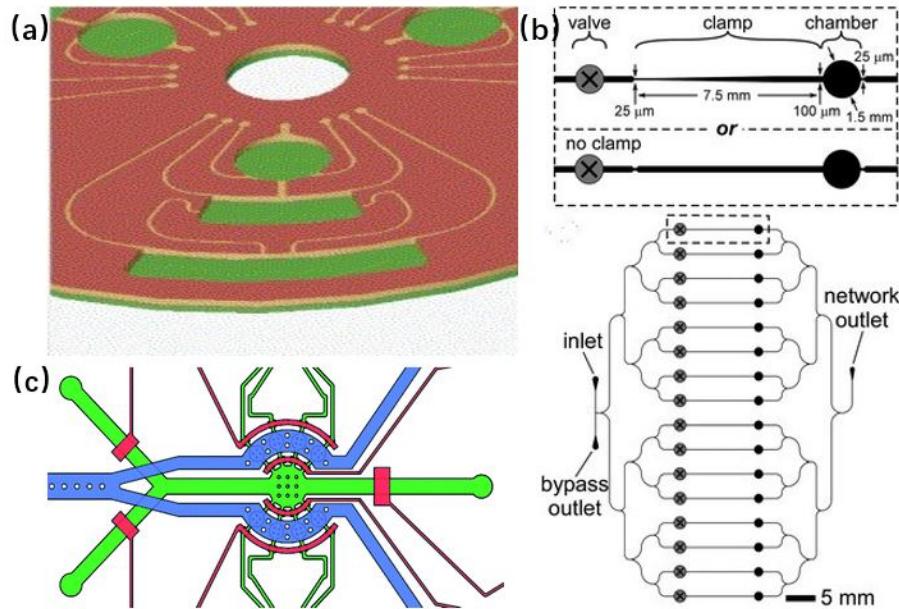


图 1-3 线虫培养芯片

Figure 1-3 Microfluidic chips for worm culture

### 1.3.1.2 线虫的固定

线虫的固定对于线虫神经元成像和线虫神经元再生<sup>[12]</sup>等需要固定线虫的实验是至关重要的。传统的线虫固定方法经常使用胶水或者麻醉剂来固定线虫，使用胶水固定线虫往往很难在短时间内恢复，而使用麻醉剂对线虫神经元可能产生潜在的影响。因此，为了实现可逆的无损伤的固定线虫，研究者提出很多基于微流控芯片固定线虫的方法。如物理化学法、结构限定法、双层阀门法、单层阀门法以及凝胶固定法。

Chokshi 等人<sup>[13]</sup>开发了一款简单的微流控芯片可以有效的固定线虫，其芯片结构如图1-4 (a)，并运用这款芯片研究了线虫的运动行为。作者比较了两种不同的固定方法，第一种方法是在上层 PDMS 通道中通入二氧化碳，利用 PDMS 材料的透气性，上层二氧化碳会向下层扩散从而达到固定线虫的目的；第二种方法是在上层通道中施加一个大的气压，使中间的 PDMS 模形变，从而可以将线虫固定。作者还通过评估线虫的运动速度来量化两种不同的固定方式对线虫产生的潜在影响，他们发现通过二氧化碳方法固定线虫比用机械的方法固定对线虫的影响较小。Saberı 等人<sup>[14]</sup>为了研究由衰老导致的线虫神经元突触形态的改变，提出了一种具有线虫同步化功能和高通量成像的微流控芯片，其芯片结构如图1-4 (b) 所示。由线虫同步化部分和线虫成像通道构成。其成像通道上有很多锥型的小通道用于捕获线虫。为了进一步固定线虫，作者在芯片上放置了干冰颗粒，通过降低温度的

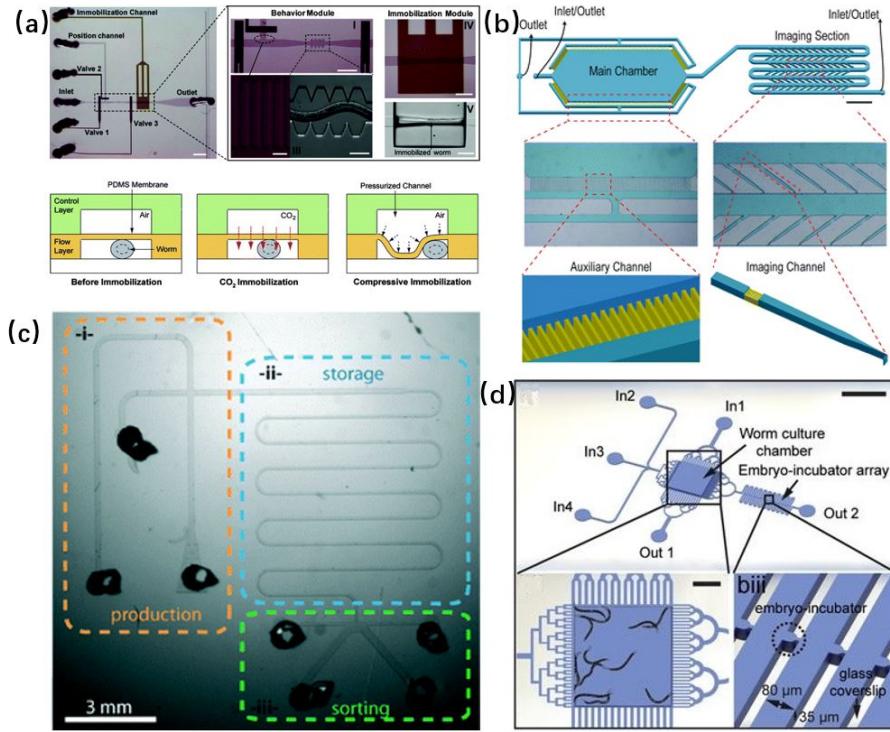


图 1-4 线虫固定芯片

Figure 1-4 Microfluidic chips for worm immobilization

方式将线虫固定。同时，为了防止二氧化碳的渗透，在芯片上表面放置了一块玻璃片。通过这款芯片，作者实现了同时对 72 条线虫的固定与成像。

以上的方法虽然能够对成虫进行固定，但由于幼虫的尺寸比成虫小且 L1 期的幼虫很容易将微通道阻塞，因此幼虫的固定是个难点。Aubry 等人<sup>[8]</sup>运用 FP127 凝胶液滴将 L1 的幼虫包裹，其芯片结构如图 1-4 (c) 所示。当液滴进入到存储单元时可以调整温度从而固定线虫，成功实现对 L1 期幼虫的操纵及成像。为了研究线虫胚胎发育及其分子机制，Cornaglia 等人<sup>[15]</sup>设计了一款可以固定线虫胚胎并对其成像的微流控芯片，其芯片结构如图 1-4 (d) 所示。其由一个线虫培养腔室和许多个微小的胚胎腔室阵列组成，研究者可以在胚胎腔室中固定线虫胚胎并且实时观察胚胎发育的整个过程，这种阵列式的腔室结构支持同时对多个线虫胚胎进行观察，具有高通量的优势。还有一些研究者利用狭窄的微通道来固定线虫<sup>[16, 17]</sup>，通过对线虫的存活率和子代数目进行统计发现这种固定方式对线虫没有产生明显影响。最近，Berger 等人<sup>[18]</sup>设计了一款单层的微流控芯片通过对阀门施加不同的压力可以控制腔室两端通道的宽度，腔室的尺寸和线虫的大小相匹配。作者运用这款芯片实现了线虫的长期固定，并且通过与自由腔室中的线虫比较，发现这种固

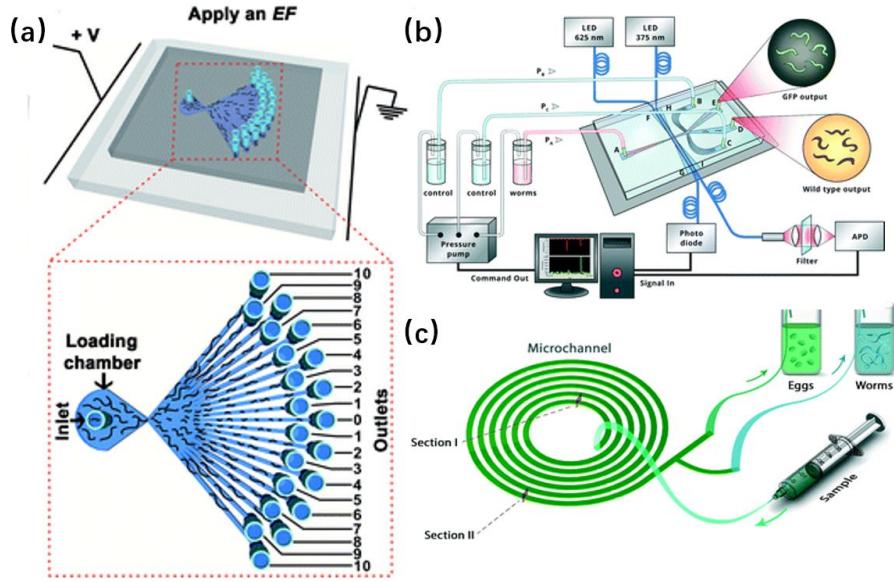


图 1-5 线虫分选芯片

Figure 1-5 Microfluidic chips for worm sorting

定方式不会对线虫产生任何影响，满足线虫的所有生理需求。

### 1.3.1.3 线虫的分选

线虫的筛选也是目前研究的热点。如对不同发育时期、不同大小的线虫进行筛选。尤其是利用线虫做为神经或发育分子学研究的模式动物时，使用正向和反向基因筛查的方式筛选新的基因是理解分子通路和蛋白功能的一种非常有效的方法。线虫基因研究比较清楚，很容易通过基因敲除等方法对线虫进行特定改造并进行相关研究。因此也需要对基因敲除的线虫进行分选。而在成百上千的线虫中筛选出特定表型的线虫是一件耗时且工作量很大的事情，运用微流控器件可以极大的缩短筛查的时间并可以很快对观察到的基因变种进行分选。线虫分选芯片的操作往往是基于线虫某种表型的特性如趋电性、行为表型、尺寸、运动能力、基因表达水平以及电生理学等特性的差异。国内刘笔锋教授课题组在 2015 年利用线虫的趋电性设计了一种单层微流控芯片<sup>[19]</sup>，可以有效的分选出不同发育阶段的线虫以及具有神经缺陷的变种，其芯片结构如图 1-5 (a) 所示。芯片的左边是线虫加载腔室，右边为许多对称的分选通道。这些分选通道宽度为 300um 高度为 120um，通道之间的夹角为 5°。不同发育阶段的线虫在电场的作用下会向不同的分选通道运动。Yan 等人<sup>[20]</sup> 开发了一种流式的线虫分选系统，并集成了光纤检测模块，其系统和芯片的结构如图 1-5 (b) 所示。片上集成的两对光纤检测模块分别用于检测

线虫是否出现和是否为荧光蛋白标记的线虫。由于整个过程不需要固定线虫，极大地提高了分选的速度。为了从混合发育的线虫中分选出线虫卵，最近，Sofela 等人<sup>[21]</sup>设计了一款螺旋线状的芯片，其芯片结构如图1-5 (c) 所示。其通道总长度为 1000um。其通道横截面高度由里到外从 220um 减小到到 160um。这种梯形横截面的设计会使流体在通道内侧形成涡流，将 5 ~ 25um 尺寸的微粒吸引到管道内侧，在螺旋管道的出口便可以根据微粒的尺寸将其分离。Casadevall<sup>[22]</sup> 等人设计了一款芯片可以对不同尺寸的线虫同步化，并且可以实现每分钟 200-1200 条线虫的分选速率。Dong<sup>[23]</sup> 等人设计了一种双层的基于 PDMS 膜形变的线虫分选芯片。通过控制上层气压使下层通道层的尺寸发生改变从而只允许特定尺寸的线虫通过这些通道进入另一个腔室。这种方法需要对特定尺寸的线虫精确控制施加气压的大小，这款芯片可以以每秒 3.5 个线虫的速率进行分选。

### 1.3.2 药物筛选与毒理实验

线虫作为一种重要的模式生物与人类的疾病基因具有约 65% 的相似性<sup>[24, 25]</sup>，因此线虫在药物筛选和毒理学研究领域经常作为一种重要的研究对象。近年来，研究者们设计了很多用于实时药物识别、筛选和毒性测试的微流控芯片。Chung 等人<sup>[26]</sup>开发了一款包含 48 个平行阵列腔室的微流控芯片，其芯片结构如图1-6 (a) 所示，每一个腔室单元的直径为 1.5mm，并且还设计了一条宽 500um 的管道用于

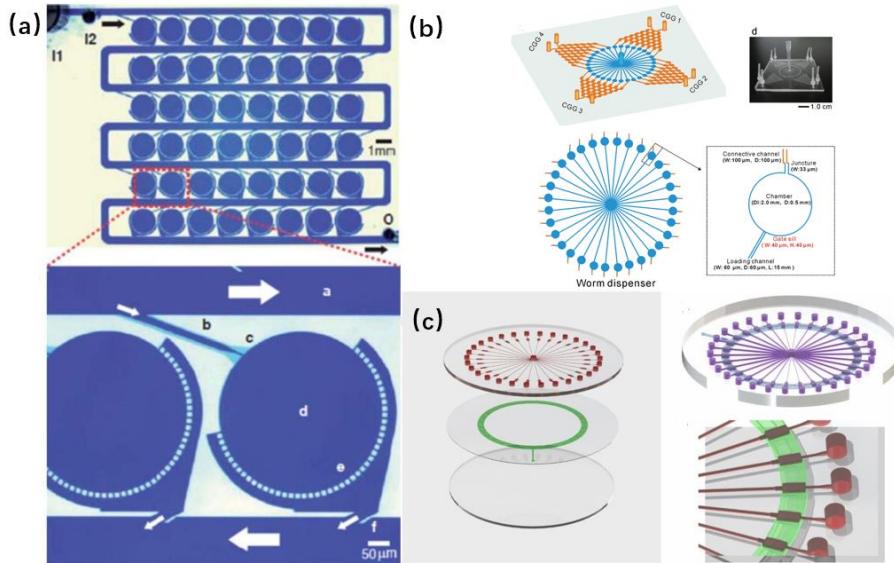


图 1-6 药物筛选线虫芯片

Figure 1-6 worm chips for drug screening

将线虫和化合物送到每个腔室。所有的腔室在视野中都是可见的，因此便于观察每个腔室中单个线虫暴露在某种化合物下的反应。Yang 等人<sup>[27]</sup>开发了一款双层的可以评估体内抗菌活性的微流控芯片，芯片结构如图1-6 (b) 所示，芯片呈现放射状四周分布着 32 个腔室，中心有一个存储的腔室，有四个“圣诞树”结构的片上浓度梯度生成器<sup>[28, 29]</sup>。利用这款芯片可以同时研究 4 种药物 32 种浓度梯度对线虫的影响。中科院大连化学物理研究所设计制作了一种三夹层的芯片，其芯片结构如图1-6 (c) 所示，该芯片由上层通道层、中间气阀控制层和底层玻璃组成。阀门层有 30 个长 2mm×宽 1mm×厚 70um 的线虫培养腔室用于线虫培养和成像，中间有一个废液池用于收集线虫培养腔室中的废液。并应用该芯片探究了高糖对线虫寿命的影响<sup>[30]</sup>。华中科技大学刘笔峰教授研究团队也利用类似结构通过实时检测相关基因的表达研究了线虫在病原体感染情况下的免疫应答<sup>[31]</sup>。

### 1.3.3 机器视觉在线虫研究中的应用

研究线虫的运动行为并对其量化在许多线虫研究中(如：神经学、遗传学和毒理学等研究中)起着十分重要的作用，通过人工观察不仅效率低下而且还会引入人为误差，如在基因筛查中需要在成百上千的线虫中发现行为异常的线虫往往需要分析几百个小时的视频，因此应用机器视觉的方法量化运动相关的表型具有重要的意义。近年来，研究者们提出很多自动化线虫图像处理的方法。Dhawan 等人<sup>[32]</sup>设计一个可以在较低的倍率下同时跟踪多个线虫的系统，但是该系统只能得到线虫运动方向等信息，无法得到线虫形态和姿势等相关特征(如体长、运动速度等)。Baek 等人<sup>[33]</sup>开发了一个可以在较高倍率下跟踪单个线虫的系统，通过控制载物台的移动可以保证线虫始终出现在视野中，通过提取 94 种不同的特征并利用决策树对特征向量进行分类可以区分出行为异常的线虫变种。在 2011 年，剑桥大学分子生物学实验室研究人员设计了一款名为 Wormtracker 的系统，该软件可以包含一套定制的软硬件系统<sup>[34]</sup>。与之前的单线虫跟踪系统相比，可以将成本下降四分之一，该系统另一个优点是能够实现对所有发育阶段的幼虫的跟踪以及游动状态下单线虫的跟踪。Swierczek 等人<sup>[35]</sup>设计了一款名为 Multi-Worm Tracker(MWT) 的软件，最多可以实现对 120 条线虫的跟踪，但只能以离线的方式对采集到的视频数据进行特征提取无法实现特征的实时提取。另外，当多个线虫的相遇并相互遮挡时，系统只会识别到一个轮廓，这样会导致跟踪的失败。

为了解决线虫跟踪过程中线虫轮廓的重叠与遮挡的问题，研究者们提出了一些基于模型的跟踪算法。Restif 等人<sup>[36]</sup>提出一种针对游动线虫的跟踪算法。其将线虫的跟踪分成两个阶段：第一个阶段是通过线虫轮廓运动的历史状态预测下一个状态线虫轮廓运动的速度，结合上一个状态下线虫轮廓的位置可以计算出当前时

刻线虫轮廓出现的位置；第二个阶段主要是调整上个阶段得到的线虫轮廓使其与当前图像中线虫的轮廓重合。Fontaine 和 Roussel 等人<sup>[37, 38]</sup> 提出了一种可以对线虫和斑马鱼进行跟踪的算法，其主要思想是通过对线虫轮廓进行参数化建模得到状态空间并通过迭代卡尔曼滤波器来预测模型参数从而实现线虫的跟踪。针对单个线虫发生自身卷曲时其轮廓识别的困难，Nagy 等人<sup>[39]</sup> 提出一种基于统计生成的方法可以识别较为复杂的姿态。

线虫的图像处理一般分为两个模块，分别是线虫的跟踪（包括线虫轮廓的分割）和线虫的特征提取。线虫的跟踪模块主要是得到线虫的轮廓，线虫的特征提取模块是利用跟踪到的线虫轮廓进行特征计算。Yemini 等人<sup>[40]</sup> 对线虫相关的行为表型进行了综述并将其分为四大类：分别为形态特征（如：体长、体宽和面积等）、姿势特征（如：线虫脊线的水平投影长度、波长以及弯曲的数量等）、运动特征（如：运动速度和运动状态等）、轨迹特征（如：线虫重心移动轨迹的范围和轨迹图等）。线虫姿态的描述一般是通过在线虫的中间脊线上等距地采样固定数量的点来定义的。这样得到线虫姿势的状态空间通常是很大的，Stephens 等人<sup>[41]</sup> 提出一种降维的方法从而可以在较低的维度空间描述线虫的姿势状态。其主要的思想是计算线虫中间脊线的曲率从而得到一个曲率向量，并对其归一化然后计算这个向量的协方差矩阵，发现其大部分的特征值为零。作者取了四个最大的特征值对应的特征向量作为四个本征向量，这四个本征向量将构成一个状态空间，任何线虫脊线曲率向量都可以投影到这个空间。由此得到四个投影长度即可描述线虫的姿势。即可以在一个四维空间中描述线虫的姿势。Restif 等人<sup>[42]</sup> 提出了一种新的计算特征的方法。其首先计算同一个线虫在不同时刻的脊线曲率向量，并将其按照时间顺序排列，由此即可得到一个曲率矩阵。通过对这个矩阵做二维离散傅里叶变换。最后基于傅里叶变换的结果计算线虫相关的特征。为了实现线虫行为数据的共享，Javer 等人<sup>[43]</sup> 提出统一的数据格式支持对视频数据以及对应的特征数据的读取和存储。并定义了一种中间数据格式表示 Worm tracker Commons Object Notation (WCON) 这样可以方便研究者组合使用不同线虫跟踪模块和特征提取模块。

## 1.4 存在的问题

尽管目前基于微流控技术的药物筛选和毒理测试取得了一定的进展，但高通量、大规模的环境化学品的毒性评价依然面临很大挑战，其主要原因是目前仍缺乏集成自动化操纵的微流控线虫分析平台。另一方面，基于微流控背景下的线虫图像处理也面临以下问题：1. 截止到目前大多数关于线虫图像处理报道并没有考虑线虫轮廓分割的问题，因为这些算法的应用场景通常是线虫和背景形成了一个很强的对比度，因此通过阈值处理即可得到线虫的前景轮廓。但线虫在微流控芯

片中的成像很难形成前景与背景对比度很强的图像，因为由 PDMS 制作的微流控芯片和线虫都是透明的，如何鲁棒地提取线虫轮廓的前景是一个难点。2. 考虑到线虫在运动的时候会出现相互遮挡并纠缠在一起以及自身卷曲等问题，线虫的跟踪是个难点。尽管目前有一些基于模型的跟踪算法可以解决一部分的线虫轮廓相互遮挡的问题，但这些方法都较为复杂且通常只能解决两个线虫相互纠缠在一起的情形，因为需要对每个线虫的轮廓运动模型建模，大大限制了其能跟踪线虫的数量。3. 由于线虫是非刚体其形态变化多种多样，线虫在多帧图像中的跟踪也是一个难点。目前在国内还没有见到关于线虫图像处理相关研究的报道。

## 1.5 研究内容

本文主要研究了微流控芯片和机器视觉在基于线虫的药物筛选和毒理测试研究方面的应用，本文的主要工作如下：

1. 本文首先制作了微流控芯片，并搭建了一个微流控芯片控制的硬件平台，可以实现线虫的简单培养并可以快速地形成多个浓度梯度，为线虫在给定化合物下的暴露创造一个精确的微环境。
2. 针对微流控背景下线虫前景轮廓分割很容易收到噪声干扰以及鲁棒性等问题，而卷积网络在特征提取方面具有很强的鲁棒性并且基于卷积网络的架构在很多领域的分割任务中取得很大的成功。本文尝试将卷积网络和条件随机场运用在线虫前景轮廓分割的任务中，结果表明，与传统的图像分割方法相比，本文提出地分割算法能够显著的改善线虫前景分割的性能，在像素分类误差指标上达到 0.11% 的分割精度。
3. 针对多线虫跟踪情况下会出现线虫轮廓相互纠缠导致跟踪丢失的问题，本文提出了一种基于深度卷积网络的网络架构。结果显示，这种方法可以有效的解析出线虫相互纠缠时单个线虫的轮廓。
4. 为了展示本文提出的软硬件平台在药物筛选和毒理测试方面的应用优势，运用该平台，本文研究了线性浓度梯度的双氧水对秀丽隐杆线虫活性的影响，并将摆动频率作为主要的监测特征。

## 1.6 论文章节安排

本文主要针对高通量药物筛选和毒理测试方面的应用，提出了一种基于微流控芯片的集片上自动化线虫操纵、线虫前景轮廓分割、线虫轮廓解析、线虫轮廓跟踪和线虫特征提取等功能的软硬件平台。本文总共分为五章：

1. 第一章为绪论，介绍了本文研究的背景及意义并对秀丽隐杆线虫的生物学

特性进行介绍。并介绍了国内外相关领域的研究进展以及目前所存在的问题。

2. 第二章主要介绍了平台的硬件部分，具体包含微流控芯片的制作以及硬件平台的搭建，最后介绍了梯度稀释芯片的验证方法，对并实验结果进行分析。
3. 第三章主要介绍了平台的线虫图像处理部分，本章首先介绍了线虫图像处理的总体流程包括线虫的前景轮廓分割、线虫轮廓的解析、线虫轮廓的跟踪和线虫特征提取。~~本章首先介绍了线虫图像处理基于深度卷积网络的线虫前景轮廓分割算法，的总体方案，然后对传统的图像分割方法进行了分类介绍，紧接着介绍了深度卷积网络的理论基础。此外，本章还介绍了线虫轮廓分割数据集的采集与标注，并重点介绍了基于条件随机场模型的深度卷积网络架构设计。最后通过实验，与传统的图像分割方法分别做了定性和定量的比较分析。~~
4. 第四章主要介绍了线虫轮廓解析算法的总体流程，紧接着介绍数据集的制作和网络结构的设计，最后介绍了网络模型的训练以及网络模型的测试，并分别从实时性、模型复杂度和准确率等指标上对两种模型进行了比较分析。
5. 第五章主要对本文的工作进行总结，并指出了下一步的研究内容。

## 第二章 微流控线虫芯片的设计及硬件平台的搭建

### 2.1 引言

在传统的药物筛选过程中，往往是通过人工的方式在 96 孔板上配置不同浓度的化合物。然后将线虫暴露在不同浓度的化合物下，观察并记录线虫在不同浓度化合物下的变化。这种人工稀释的方法不仅存在样品消耗大、通量低和操作繁琐等缺点，而且也不利于观察。近年来，随着微流控技术的发展，一些研究者们提出了基于微流控技术的片上浓度稀释的方法。片上浓度梯度稀释的方法不仅使反应体系减小十倍甚至百倍，而且极大的减少了试剂的消耗，节约了实验成本。虽然已经有“圣诞树”结构的被动式梯度形成芯片的报道，但存在样品消耗大以及需要精确的流阻设计和流速调节。为了研究多种化合物的复合对线虫活性的影响，本文设计了一种基于振荡的线性梯度稀释的微流控芯片。且只需要一个气源即可通过振荡的方式完整样品的快速稀释，并通过染料实验和荧光实验证本文提出的线性浓度梯度芯片的可行性。

### 2.2 双层线虫梯度芯片的设计及制作

#### 2.2.1 线虫梯度芯片的设计

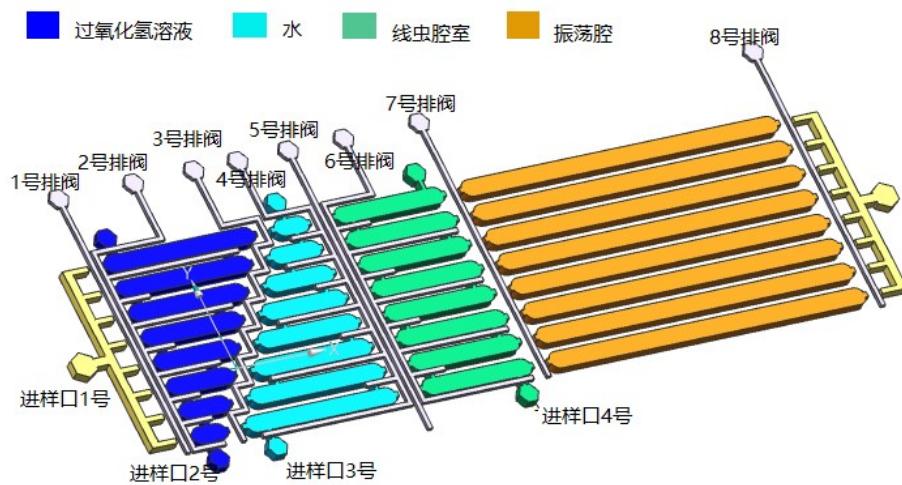


图 2-1 线性梯度稀释芯片结构

Figure 2-1 Structure of Linear gradient dilution chip

线虫梯度芯片的设计如图2-1所示，由两层 PDMS 管道组成。上层为流道层，下层为阀门层。芯片包含八个阀门和四个进样口。阀门分别控制每列之间的隔绝和开启以及每列腔室之间的相互隔绝和开启。其中 3、5、7 号阀门控制 4 列每列之间的隔绝，而 2、4、6 号阀门分别控制 1、2、3 列里 9 排腔室之间的隔绝，8 个阀门将整个芯片分为  $9 \times 4$  个独立的腔室。4 列腔室中第一列腔室的长度依次线性递减，第二列腔室的长度依次线性增加，前两列腔室的长度总和为 2mm (两腔室长度的比例从上到下依次为 9 : 1, 8 : 2, …, 1 : 9)。第三列和第四列腔室的长度保持恒定分别为 1mm 和 3mm。所有腔室的宽度和高度分别为 200um 和 80um。

## 2.2.2 实验材料与仪器

表 2-1 实验试剂与耗材

Table 2-1 Experimental reagents and consumables

光刻胶 SU-83050	美国 Microchem 公司
光刻胶 AZ4903	德国 Merck Performance Materials 公司
聚二甲基硅氧烷 PDMS( poly-dimethylsiloxane, RTV 615)	美国 Momentive Performance Materials 公司
氟化液 FC-40	北京伊诺凯科技有限公司
FA1004 分析电子天平	上海良平仪器仪表有限公司
RC8 型匀胶机	美国 Karl Suss 公司
DZF-6020 型真空干燥箱	上海新苗医疗器械制造有限公司
ZXZ-2 型旋片式真空泵	浙江谭氏真空设备有限公司
SMZ-168 型体式显微镜	中国麦克奥迪公司

## 2.2.3 芯片模具加工工艺

### 2.2.3.1 阀门层模具制作

- a) 按照2.2.1小节的结构设计使用 AutoCAD 绘图软件绘制阀门层的结构图案并制作掩模版。
- b) 在涂胶之前，需先将 3 英寸的单抛硅片放在  $180^{\circ}\text{C}$  的烘箱中 2 小时，然后用氧等离子体去胶机处理 60 秒。
- c) 旋涂光刻胶 AZ4903 并用 200rpm 的转速预转 8 秒，然后再用 1000rpm 的转速甩胶 25 秒得到 22um 厚的胶。

- d)** 将硅片放入烘箱中，逐渐升温至  $50^{\circ}\text{C}$ ，30 分钟后再将温度升高至  $90^{\circ}\text{C}$ ，90 分钟后再将其取出。
- e)** 经过紫外曝光后再显影。
- f)** 为了去除硅片表面的水汽，将其放入  $80^{\circ}\text{C}$  的烘箱中 1 小时，最后使用甲基三氯硅烷处理硅片表面 5 分钟。

### 2.2.3.2 流道层模具制作

由于本文中芯片的管道和腔室的高度被设计成不同的尺寸，且腔室的高度要比管道的高度要高，所以需要经过两次光刻才能完成流道层模具的制作，第一次光刻采用正胶制作管道结构，第二次光刻采用负胶制作腔室结构。

- a)** 使用 AutoCAD 绘图软件分别绘制芯片管道和芯片腔室的结构并打印制作掩模。
- b)** 在涂胶之前，需先将 3 英寸的单抛硅片放在  $180^{\circ}\text{C}$  的烘箱中 2 小时，然后用氧等离子体去胶机处理 60 秒。
- c)** 旋涂光刻胶 AZ4903 并用 200rpm 的转速预转 8 秒，然后再用 1000rpm 的转速旋涂 25 秒得到 22um 厚的胶。
- d)** 将硅片放入烘箱中，逐渐升温至  $50^{\circ}\text{C}$ ，30 分钟后再将温度升高至  $90^{\circ}\text{C}$ ，90 分钟后再将其取出。
- e)** 经过紫外曝光后再显影。
- f)** 将硅片放在热板上分别升高温度分别在  $65^{\circ}\text{C}$ 、 $95^{\circ}\text{C}$ 、 $120^{\circ}\text{C}$  和  $190^{\circ}\text{C}$  分别停留 5min、5min、40min 和 60min。
- g)** 旋涂负胶 SU-8 3050 并用 500rpm 的转速预转 8 秒，然后再用 1500rpm 的转速旋涂 30 秒得到 80um 厚的胶。
- h)** 将硅片放在  $95^{\circ}\text{C}$  的烘箱中恒温 40min。
- i)** 经紫外曝光后将硅片放入  $95^{\circ}\text{C}$  的烘箱中恒温 60min。
- j)** 用对应的显影液对硅片显影，并用异丙酮溶液漂洗，以除去残留的显影液和试剂残留并用氮气将硅片吹干。
- k)** 为了除去水汽，将硅片放入  $80^{\circ}\text{C}$  的烘箱中恒温 60min，再用甲基硅氧烷处理 5min。

### 2.2.4 线虫梯度芯片的制作

制作好芯片模具后，就可以利用芯片模具进行微流控芯片的制作，下面将介绍双层微流控芯片的制作流程。经过以下流程最终完成的芯片其实物图如图2-2所示，为了更好的显示芯片结构，芯片的四列腔室都被打入不同颜色的染料。

- a) 将流道层硅片的背面用胶粘在一次性培养皿的底部，起到一个固定的作用。
- b) 配制 PDMS：按照 20: 1 的比例，取 20 克 RTV 615 (A 液) 和 1 克固化剂 (B 液) 混合搅拌均匀，按照 5: 1 的比例，取 20 克 RTV 615 (A 液) 和 4 克固化剂 (B 液) 混合搅拌均匀，并将两种比例的混合液放入真空皿中抽真空以排出液体中的气泡，不断地抽气放气直到液体呈现澄清状。
- c) 将 5: 1 的 PDMS 混合液倒入流道层培养皿中，PDMS 混合液的厚度大约 5mm 左右。
- d) 将阀门层硅片放在甩胶机吸盘的中心位置，并用真空吸住硅片使其固定，将 20: 1 的 PDMS 混合液倾倒在硅片上，用 400rpm 的转速预转 20 秒后用 1800rpm 的转速转 60 秒即可得到厚约 40um 的 PDMS 层。
- e) 将阀门层模具和流道层模具一起放入 80°C 的烘箱中恒温 30min。
- f) 用手术刀将培养皿中的 PDMS 层沿着硅片的边沿切下并取出。再用铲刀沿着芯片上的图案将其切成长方形。
- g) 在显微镜下将上一步切下的流道层芯片与阀门层对准并粘合在一起。
- h) 将其放入 75°C 的烘箱中恒温 5 个小时进行高温键合，然后将其取出放在无尘纸上进行常温冷却。
- i) 再用打孔器将芯片的进样口、出样口和阀门入口位置打孔。
- j) 将芯片和干净的玻璃片一起放入氧等离子体去胶机中处理 40 秒后，将芯片和玻璃贴合在一起并放入 80°C 的烘箱中恒温 2 小时。

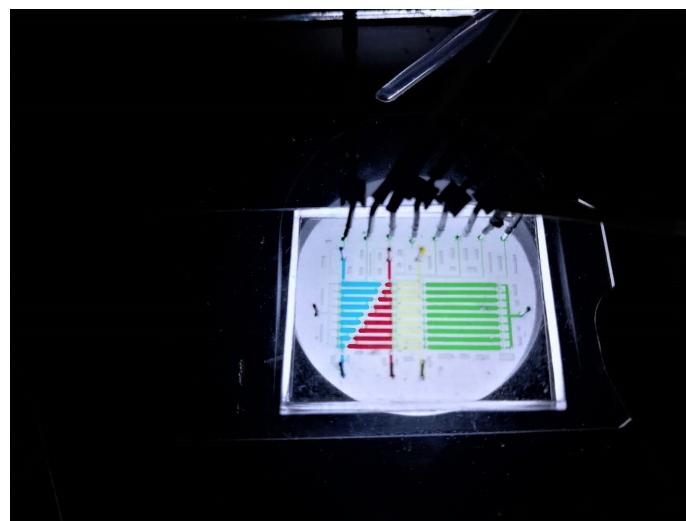


图 2-2 线性梯度稀释芯片实物图

Figure 2-2 fabricated linear gradient dilution chip

### 2.2.5 振荡的理论机制

在微流控芯片中，流体的雷诺系数  $Re$  变得很小，流体以层流的方法流动，这使得微管道中的不同液体的混合变得非常缓慢。我们课题组提出了一种只需一个压力源的基于流体振荡的混合器<sup>[44]</sup>，振荡的时候需要在管道的一端构造一个封闭的空腔环境。当需要混合前两列腔室中的液体时，需要将第三列作为储气腔，将第三列腔室右边的阀门关闭即可形成封闭的储气腔。当需要混合前三列腔室中的液体时可以将第四列腔室作为储气腔。振荡的时候将九排腔室之间的阀门关闭，对其中一排腔室的受力分析可等效为图2-3，其中  $P_{atm}$ 、 $P_{push}$  和  $P_{air}$  分别表示大气压、外部施加的压强与大气压的差和储气腔中的气压。储气腔由于封闭了一段空气，当开始施加外部压力时，此时外部气压大于储气腔内的气压，会推动液体往储气腔运动，储气腔内的气体由于受到液体的挤压，体积会收缩且内部压强开始增大；当外部压力撤掉时，由于储气腔中的压强大于外部的压强，液体会向反方向运动，这时储气腔的体积会增大压强会变小。当压强下降到等于外部压强时，液体会停止向前运动，继续施加压力，液体又会重复上述运动，这样液体会在周期性气压的作用下来回振荡，类似于宏观的混合吹打从而完成不同溶质的混合。

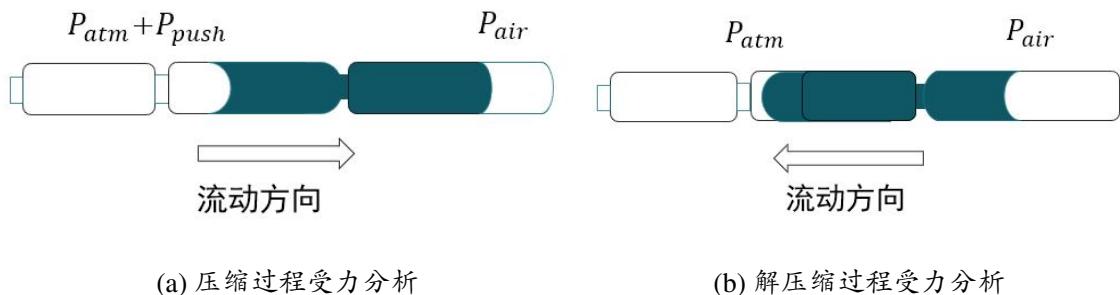


图 2-3 液体振荡过程受力分析

Figure 2-3 Stress analysis of liquid oscillation process

### 2.2.6 染料与荧光实验

为验证振荡混合效果，我们使用染料及荧光进行验证。如图2-4a所示，将第一列腔室打入红色染料，第二列腔室打入水，将第三列腔室作为振荡腔。在 1 号进样口施加一个周期为 250ms 压力值为 0.05Mpa 的气压，经过 7 分钟的振荡，染料已经充分的混合均匀。染料和水完全混合后的效果如图2-4b所示，从上到下染料的颜色依次变淡。如不加振荡，两列液体以自由扩散方式混合，需要 2~3 个小时，可以看出，振荡方法对快速混合形成梯度有较好的效果，大大加快了不同液

体的混合速率。为了定量的验证梯度形成的准确性，我们将第一列腔室中的染料替代为浓度为  $0.1\text{g/L}$  的荧光素钠溶液 ( $\text{pH}=7$ )，最终得到  $0.01\sim 0.09\text{g/L}$  的线性浓度梯度。图2–6为荧光素钠浓度与荧光强度之间的关系。图中蓝色的线表示线性拟合的结果，可以看出拟合结果的线性度较好。从而验证了基于微振荡原理的大规模梯度稀释是可行的，这种方式能够显著加快混合的速度并且只需要一个压力源即可完成振荡混合。为了验证这种振荡对线虫并无不良后果，我们将线虫打入第三列腔室，并在该参数下振荡，观察 2 个小时发现线虫的摆动频率并没有出现太大波动。

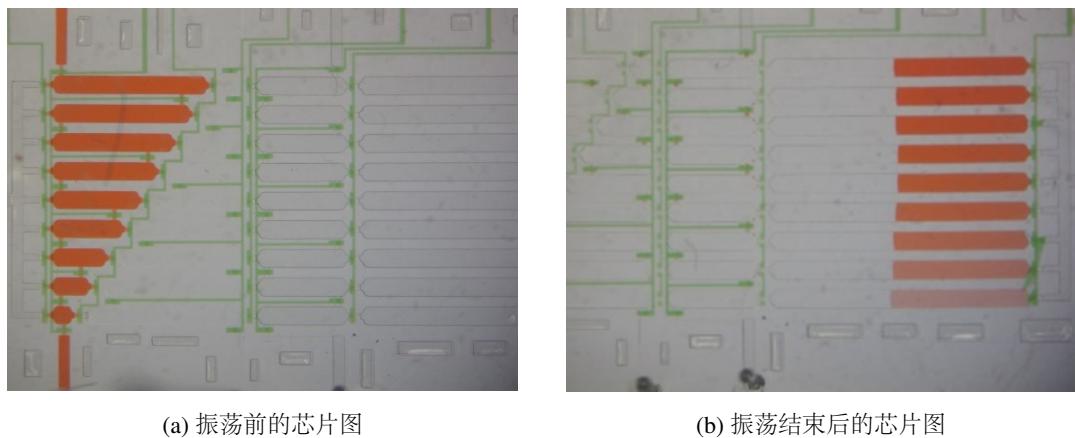


图 2–4 染料实验结果

Figure 2–4 Dye experiment results

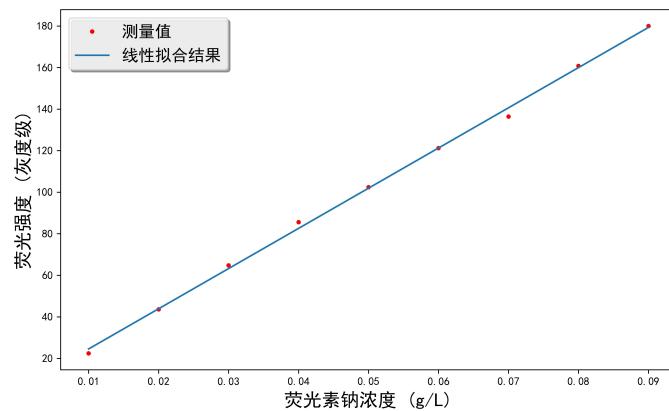


图 2–5 荧光实验结果

Figure 2–5 Fluorescence experiment results

## 2.3 单层侧向阀门线虫培养芯片的设计及制作

### 2.3.1 单层阀门线虫培养芯片的设计

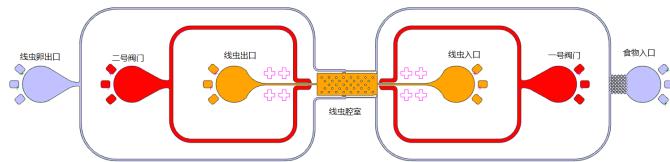


图 2-6 单层阀门线虫培养芯片

Figure 2-6 Single layer lateral valve chip for worm culture

### 2.3.2 单层阀门芯片模具的制作

下面开始介绍单层阀门芯片模具的制作工艺流程。

- a) 根据芯片结构的尺寸设计运用 AutoCAD 绘图软件绘制芯片的二维结构并制作掩模版。
- b) 为了使光刻胶和硅片更容易结合，必须充分去除硅片的水分。将单面抛光的 3 寸硅片放进  $180^{\circ}\text{C}$  烘箱中恒温  $2 \sim 3$  个小时。
- c) 将硅片从烘箱中取出，待其恢复到室温后，用氮气将硅片表面的杂质吹掉使硅片表面保持干净。用光刻胶 AZ-4903 倒在硅片的中心处，使其自然铺开慢慢覆盖硅片表面。然后  $5000\text{rpm}$  的转速旋涂  $30\text{s}$  得到胶层厚度大约为  $1\mu\text{m}$ 。
- d) 为了去除光刻胶中的有机溶剂并使光刻胶固定需要经过前烘，将涂有光刻胶的硅片放入烘箱中。将烘箱的温度缓慢升至  $65^{\circ}\text{C}$ ，恒温  $30\text{min}$ 。然后继续将温度升至  $90^{\circ}\text{C}$ ，再恒温  $30\text{min}$ ，最后慢慢将烘箱温度降至室温。
- e) 将硅片放入光刻机中，曝光前，先打开光刻机汞灯预热  $10\text{min}$  分钟。将掩模版放在硅片上面并中心对齐，根据光刻机的厚度调整好曝光时间，经过曝光处理后，将可掩模上的图案转移到光刻胶上。
- f) 将经过曝光处理的硅片放入烘箱中逐渐升温至  $65^{\circ}\text{C}$  并恒温  $15\text{min}$ ，然后再逐渐升温至  $95^{\circ}\text{C}$  恒温  $40\text{min}$ ，最后慢慢将温度降到室温。
- g) 经过后烘步骤，再用对应的显影液对硅片进行显影，得到清晰的图形后再用氮气将硅片表面吹干。
- h) 将硅片放入深硅刻蚀机中，进行干法刻蚀。光刻胶覆盖的区域将不会被刻蚀，从而可以将掩模板上的芯片图案转移到硅片表面。注意根据腔室的高度

度，控制好刻蚀的厚度。

- i) 为了使 PDMS 芯片的制作过程中，PDMS 层容易从模具上揭下来，需要对硅片表面进行硅烷化处理。将硅片放入一次性的培养皿中，在硅片旁的滤纸上滴一滴硅烷化试剂，通过蒸发沉淀的方式完成硅烷化处理。

### 2.3.3 单层阀门芯片的制作

将 PDMS 浇筑在芯片的模具上待其固化，即可完成

- a) 将硅片模具固定在一次性的培养皿的底部。
- b) 配置 PDMS 混合液：按照 10: 1 的比例，取 20 克的 PDMS(Sylgard 184)A 液和 2 克的固化剂 (B 液) 混合搅拌均匀，并将两种比例的混合液放入真空皿中抽真空以排出液体中的气泡，不断地抽气放气直到液体呈现澄清状。
- c) 将 PDMS 浇筑在放有硅片模具的培养皿中，浇筑厚度约为 5mm，重新放回真空皿中抽出芯片表面的气泡。
- d) 将培养皿放入 75°C 的烘箱中恒温一个小时使 PDMS 得以固化。
- e) 从烘箱中取出芯片模具，待其自然冷却，用手术刀将 PDMS 层从模具上揭下来。按照芯片上的图案用 PDMS 切成方形，并在所有进口、出口处打孔。
- f) 将芯片和干净的玻璃片一起放入氧等离子体去胶机中处理 40 秒后，将芯片和玻璃贴合在一起并放入 80°C 的烘箱中恒温 2 小时，加强键合效果。

### 2.3.4 单层侧向阀门线虫固定芯片的测试

## 2.4 阀门控制系统搭建

目前，双层的微流控芯片一般由下层的阀门层和上层的流道层组成。阀门层的 PDMS 薄膜通常很薄，当向气阀通道施加一定的气压(大约 30psig)，阀门层的薄膜会向上顶起。由于上层的流道和下层的气阀通道是上下交叉的，通过控制向气阀通道施加气压，便可以控制上层流道的通断。另一方面，线虫的压力进样和芯片的振荡都需要对进样口进行气压控制。基于以上需求，本课题组开发了自动化阀门控制系统，图2-7为该系统的连接示意图。整个系统由五个模块构成，我们采用了 750W-18L 小型空压机作为气泵模块为整个系统提供气压输出，将 Arduino 作为微控制器控制电磁阀的输出。Arduino 由运行在电脑上的上位机程序控制。

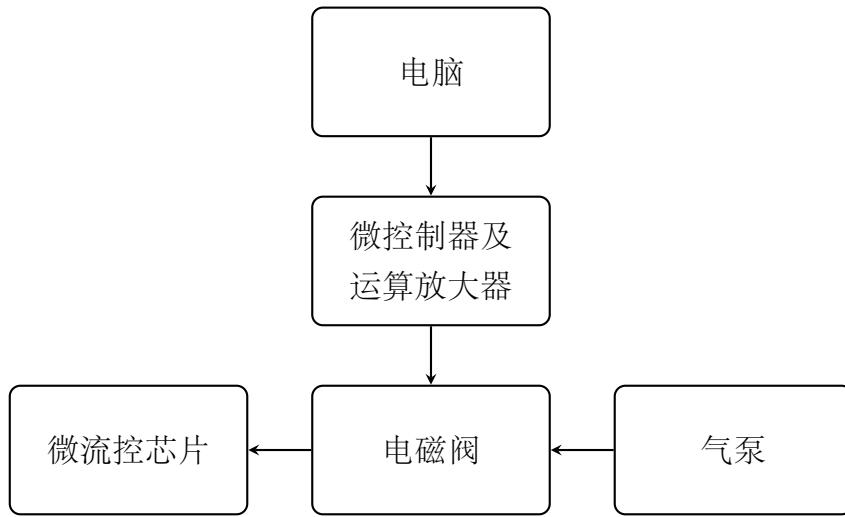


图 2-7 自动化阀门控制系统

Figure 2-7 Automated valve control system

## 2.5 高速视频采集程序的设计

线虫视频的高速采集对线虫摆动频率的估计是至关重要的一步。根据采样定理，采集的视频帧率至少是线虫摆动频率的两倍。除了采样频率的要求外，视频的分辨率等因素同样影响后续线虫图像处理的难度。基于以上考虑，本文选取鑫图公司的 DigiRetina 16 COMS 相机作为线虫视频的采集模块。其可以通过 USB3.0 高速传输接口与电脑进行高速图像数据传输，同时该相机在放大倍率 20 倍以下的显微成像中，也可以获得分辨率优异的显微图像，另外其内置两个 FPGA 双核处理器分别用于高清图像处理和图像输出控制。该相机较好满足了本文对线虫视频采集模块的要求。为了提高视频采集的帧率，本文基于该相机的 SDK，开发基于多线程的视频采集程序。

对 DigiRetina 16 相机设备的操作通常分为 7 个步骤，其操作流程如图2-8所示。设备初始化阶段主要负责和相机驱动通信并对其初始化，在图像采集之前，需要在内存中开辟内存空间用于保存采集的图像数据，所需内存的大小与图像的分辨率有关。DigiRetina 16 相机支持多种分辨率的图像采集，分辨率可以在打开设备后进行设置。在结束采集后，需要关闭设备并释放内存空间。

一帧图像从采集到被处理共分为 4 步骤，如图2-9所示，各部分的时间开销分别用  $T_1, T_2, T_3, T_4$  表示，下面分别对各部分时间开销进行详细分析：

**时间开销 T1：** 曝光时间和图像分辨率影响这部分的时间开销，增加曝光时间或提高图像采集的分辨率都会使这部分的时间开销变大。

**时间开销 T2：** 相机驱动程序负责将相机缓存中的图像数据传送到电脑上的一

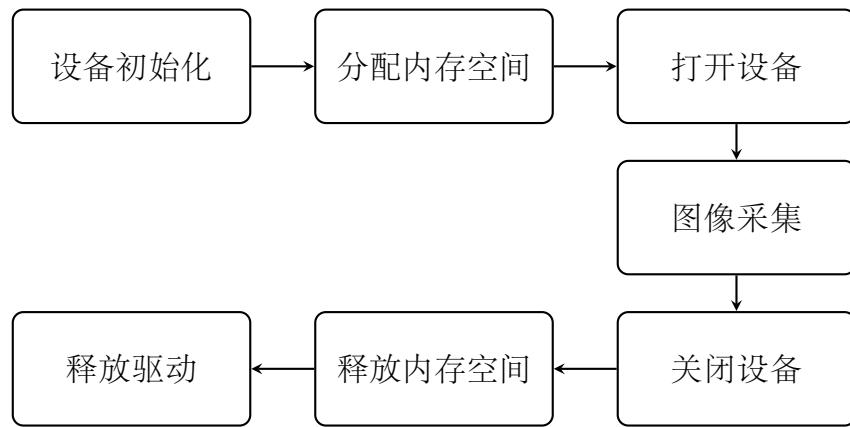


图 2-8 相机设备的操作流程

Figure 2-8 Camera device operation flow

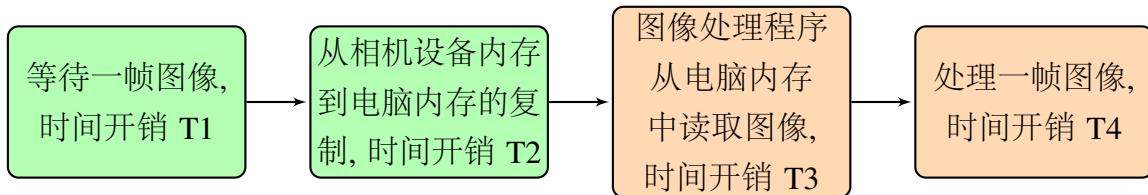


图 2-9 从图像采集到图像处理过程的时间开销分析

Figure 2-9 Time consumption analysis from image acquisition to image processing

块内存区域。这个内存块在相机初始化时开辟，释放相机设备时自动释放，图像采集的分辨率、USB 线的长度会影响时间开销。

**时间开销 T3：**图像处理程序从上一步的内存区域读取图像数据。图像分辨率、内存拷贝数据的速度等会影响这部分的时间开销。

**时间开销 T4：**处理一帧图像的时间对不同的图像处理程序而言是不同，这部分的时间开销与具体的图像处理算法有关。对于高速视频采集任务而言，这里的时间开销指的是写入硬盘时间。

从一帧图像的采集到处理也可以分成图像采集阶段(对应图2-9前两个部分)和图像处理阶段(对应图2-9后两个部分)。如图2-10 (b) 所示，为了加速视频采集的帧率，本文采用了两个独立的线程分别负责图像的采集和将图像写入硬盘，这样可以将这两个任务并行，一帧图像从采集到写入硬盘平均所消耗的时间取决于图像采集和硬盘写入两者所消耗时间的最小值，公式2-1表示多线程模式下视频采集帧率。而在单线程模式下，这两个任务是顺序执行的，一帧图像从采集到写入硬盘平均所消耗的时间取决于图像采集和硬盘写入两者消耗时间之和，公式2-2表示单线程模式下视频采集帧率。多线程视频采集相较于单线程视频采集的加速比由公

式2-3表示。

$$\text{多线程下视频采集帧率} = \frac{1}{\min(T_1 + T_2, T_3 + T_4)} \quad (2-1)$$

$$\text{单线程下视频采集帧率} = \frac{1}{T_1 + T_2 + T_3 + T_4} \quad (2-2)$$

$$\text{加速比} = \frac{T_1 + T_2 + T_3 + T_4}{\min(T_1 + T_2, T_3 + T_4)} \quad (2-3)$$

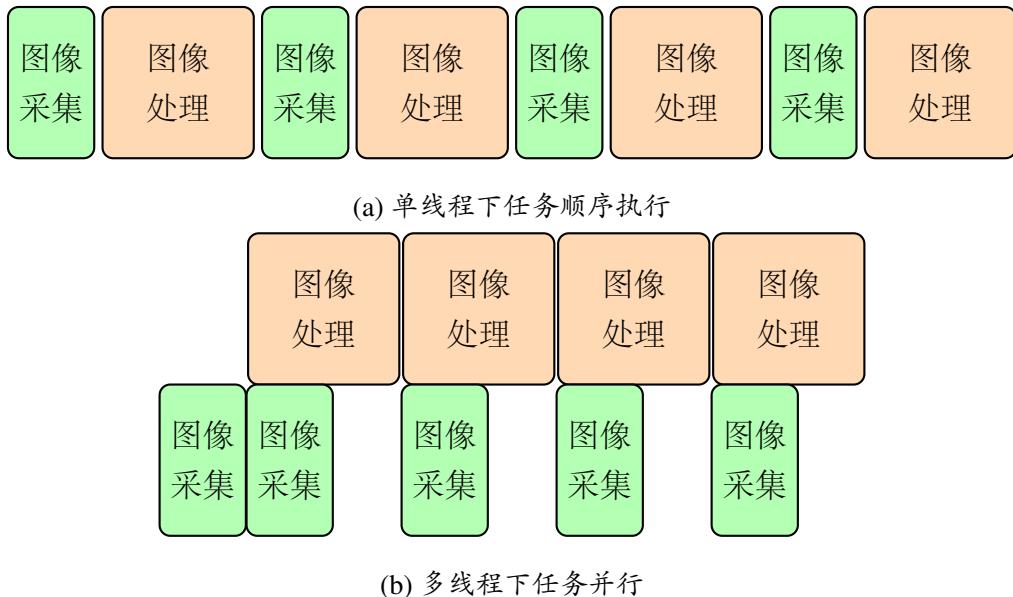


图 2-10 单线程和多线程模式下时间开销对比

Figure 2-10 Time overhead comparison in single-threaded and multi-threaded mode

为研究不同采集分辨率对视频采集帧率的影响，本文测试在不同采集分辨率情况下，各部分时间开销的变化，表2-2显示了本文的测试结果。结果显示分辨率提高对各部分的时间开销均有显著影响，在同一分辨率下，多线程视频采集相较于单线程视频采集其帧率均有大幅提高。且随着采集分辨率的提高，加速效果越明显。

## 2.6 本章小结

本章的主要工作是为本文提出的药物筛选平台搭建了一个硬件平台。首先，介绍了基于振荡原理的线性梯度稀释芯片的结构设计。本章还介绍了微流控芯片的制作工艺。然后介绍了振荡混合的理论机制，并在最后通过染料实验和荧光实验

表 2-2 不同分辨率对视频采集帧率的影响

Table 2-2 The effect of different resolutions on the video capture frame rate

分辨率	4608 × 3456	2304 × 1728	1600 × 1200
T1	192ms	38.07ms	39.12ms
T2	7.7ms	1.84ms	0.772ms
T3	7.9ms	2.28ms	0.773ms
T4	272.4ms	22.89ms	16.81ms
多线程帧率	3.5frames/sec	25.0frames/sec	25.0frames/sec
单线程帧率	2.0frames/sec	15.0frames/sec	17.4frames/sec
加速比	1.75	1.67	1.44

证明了线性梯度稀释芯片设计的合理性，实验结果显示基于振荡的混合方式能够有效的缩短不同液体的混合时间。

### 第三章 自动化线虫视频分析与特征提取

#### 3.1 引言

在基于线虫的毒理实验中，通常需要观察线虫在给药的过程中其生理特征的变化，如在基于线虫的神经毒性实验中需要对线虫的存活率、身体弯曲频率、头部摆动频率等特征进行统计。在基于线虫的发育毒性实验中，需要对线虫的身体长度、形状、产卵器形态发育等特征进行统计。在基于线虫的生殖毒性实验中需要对线虫的胚胎发育、性腺形态发育、后代数目进行统计观察。通过人工观察的方式不仅会引入误差，而且效率低下。由于实验数据分析上存在瓶颈，严重限制了实验的规模。因此，线虫的自动化视频分析和特征提取对于大规模的毒理实验和药物筛选是至关重要的。

针对线虫视频特征提取任务的特点，本文采用了“前景轮廓分割——轮廓解析——轮廓跟踪——特征提取”的技术路线。其中前景轮廓分割和轮廓解析是整个技术路线中关键的一步，其直接影响到能否对线虫进行有效的跟踪以及特征提取等后续步骤。线虫轮廓分割的不完整会导致跟踪的丢失率显著上升。为了提高线虫轮廓分割的精度，本章将深度卷积网络应用在线虫前景轮廓分割和轮廓解析任务中。并与传统的分割算法做了详细的比较分析，发现基于深度卷积网络的分割算法能够显著的改善线虫前景轮廓分割的效果。为了实现对线虫轮廓的跟踪，本章提出了一种基于最近邻搜索的跟踪方法，在线虫轮廓分割较为完整的情况下，能够实现对线虫的鲁棒跟踪。最后，介绍了线虫体长和摆动频率等特征的计算方法。

#### 3.2 线虫图像处理总体方案介绍

图3-1显示了线虫视频特征提取的技术路线流程，共分成四个阶段。第一个阶段为线虫前景轮廓的分割，从采集的视频中读取一帧图像，并将线虫轮廓覆盖的区域定义为前景，剩下的区域视为背景。利用图像分割算法进行前景背景分割后得到一幅二值化的图像，其中1表示前景0表示背景。如果是单线虫的跟踪，则第一个阶段的输出则为线虫的轮廓。但考虑到多线虫跟踪过程中多个线虫的轮廓会出现相交甚至纠缠在一起，导致因无法区分单个线虫轮廓从而跟踪丢失。第二个阶段的任务是对多个纠缠在一起的线虫轮廓进行解析，从而得到单个线虫的轮廓。第三个阶段是线虫轮廓的跟踪，利用线虫轮廓的重心、面积等信息找出相邻两帧图像之间线虫轮廓的对应关系。第四个阶段为线虫特征计算，利用上一阶段跟踪

的线虫轮廓计算线虫体长、运动速度和摆动频率等生理特征。本章的剩余部分将依次对各个部分进行介绍。

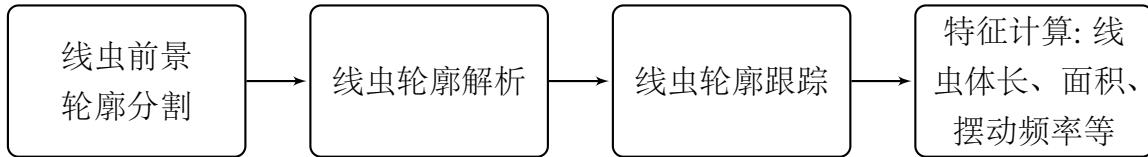


图 3-1 线虫图像处理总体流程图

Figure 3-1 The flow chart of *C.elegans* image processing

### 3.3 卷积神经网络介绍

从 1956 年正式提出人工智能学科以来，人工智能领域的研究经历了 60 年的发展。期间，不同学科背景的学者们对人工智能的实现，提出了不同的观点并由此产生了不同的学术流派。其中对人工智能领域影响较大的主要有符号主义 (Symbolism)、连接主义 (Connectionism) 和进化主义 (Evolutionism) 或控制论学派 (Cyberneticsism) 等。目前，在诸多领域取得重大突破的深度学习 (Deep learning) 技术是连接主义学派的典型代表性技术。其主要受神经科学的启发，大脑中大量的神经元形成复杂的连接，信号在神经元之间传递。人工神经网络 (Artificial Neural Network, ANN) 由大量相互连接的节点构成，每个节点代表一个特定激活函数的输出，每两个节点之间的连接都有一个权重。通常，神经网络模型都是由多层网络堆叠而成，前一层网络的输出作为后一层网络的输入。从网络的输出来看，整个网络的输出相当于由许多张量函数嵌套构成的复合函数，网络模型参数的训练依赖的误差反向传播算法其实质是复合函数的链式求导法则。

卷积网络 (Convolutional Neural Network, CNN) 是人工神经网络模型中的一种，由 Yan Lecun 于 1989 年提出并将其用于手写数字识别<sup>[45]</sup>，卷积操作类似于图像处理中的空间滤波，与之不同的是滤波核作为模型参数是通过学习得到的。一个典型的卷积网络架构如图3-2所示，通常由卷积层，池化层，全连接层等构成。为了使模型具有更好的泛化性能，通常在不同的层之间还会加入 Dropout 层和 Batch Normalization 层。自 2006 年以来，研究者们不断对卷积网络进行改进提出了很多性能优异的网络架构 (如：AlexNet 网络<sup>[46]</sup>、VGG 网络<sup>[47]</sup>、GoogLeNet 网络<sup>[48]</sup> 和 ResNet 网络<sup>[49]</sup>)。使卷积网络在许多计算机视觉任务中大放异彩。

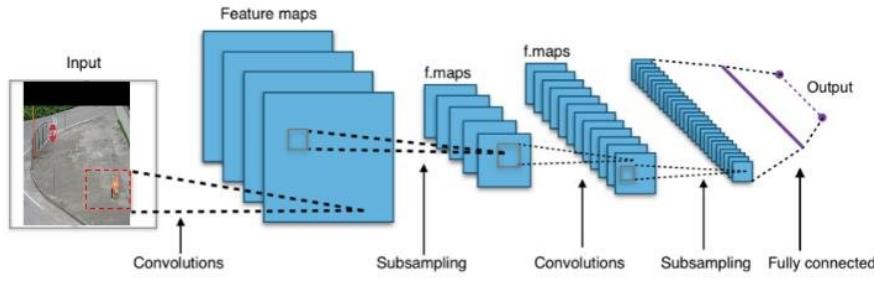


图 3-2 卷积网络的经典结构

Figure 3-2 A typical architecture of CNN

### 3.3.1 卷积层

卷积层是卷积网络中主要的连接层，由于其引入权值共享的机制，与全连接的方式相比将网络的模型参数减小了几个数量级，且对输入图像的缩放和旋转等变形具有高度不变性。如图3-3所示，卷积操作可以理解为卷积核在输入张量上的滑动。输出张量的尺寸大小由卷积核的大小、填充的大小、步长和输入张量的大小决定。公式3-1表示了这一关系，其中  $o$  和  $i$  表示输出张量和输入张量在某一维度的大小， $p$  和  $s$  分别表示填充的大小和卷积步长， $k$  表示卷积核的大小， $\lfloor \cdot \rfloor$  表示向下取整。图3-3中  $i = 6$ ,  $p = 1$ ,  $k = 3$ ,  $s = 2$  根据公式3-1 可以得到  $o = 3$ 。

$$o = \lfloor \frac{i + 2p - k}{s} \rfloor + 1 \quad (3-1)$$

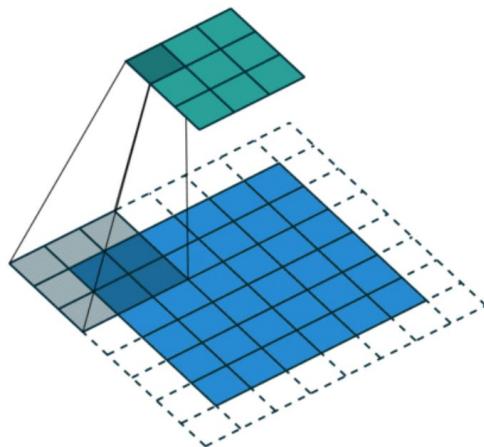


图 3-3 卷积操作示意图

Figure 3-3 Convolution operation diagram

### 3.3.2 池化层

池化层是一种对输入张量进行降采样的操作，由于池化操作是在输入张量的每一个通道上进行的，所以不会改变通道的数目。池化操作可以大致分为最大值池化、均值池化和随机池化三种，这三种池化操作都是通过对输入张量一个邻域内的像素值进行聚合统计实现的。因为部分像素的改变并不会对聚合统计的结果造成太大的差异，因此池化操作具有一定的尺度不变性。如图3-4是三种池化操作的示意图，最大值池化求领域内特征点的最大值作为统计输出，均值池化求领域内特征点的平均值作为统计输出，随机池化将领域内特征点按照其数值大小赋予概率，然后按概率采样得到结果作为统计输出。

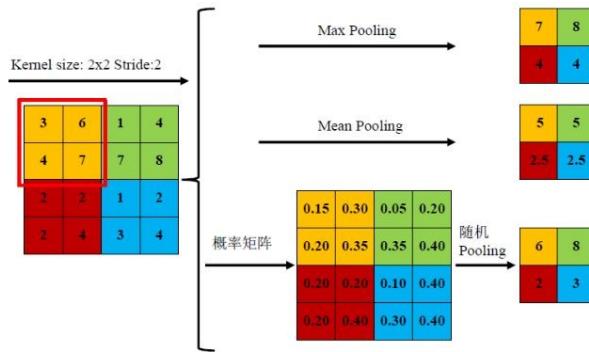


图 3-4 三种池化操作示意图

Figure 3-4 Three pooling operations diagram

### 3.3.3 Dropout 层

为了解决神经网络过拟合的问题，Srivastava 和 Hinton 等人<sup>[50]</sup>于 2014 年提出 dropout 方法，有效的提高了网络的泛化能力。如图3-5所示，在网络训练阶段，对每一个神经元以一定的概率丢弃。从而每次更新网络权值时，网络的连接都是随机的，相当于对采样到的网络进行更新。在网络测试阶段，为了得到所有随机连接网络的平均输出，这个阶段网络使用全连接的方式并且将网络的权值以一定的比例缩放。大量的实验表明，采用 Dropout 的网络具有很好的泛化能力且能够加速网络的收敛。

### 3.3.4 Batch Normalization 层

批量标准化 (Batch Normalization,BN) 是深度学习领域一种非常重要的正则化的方法<sup>[51]</sup>，有效的解决了内部方差偏移 (Internal Covariate Shift) 的问题。内部方

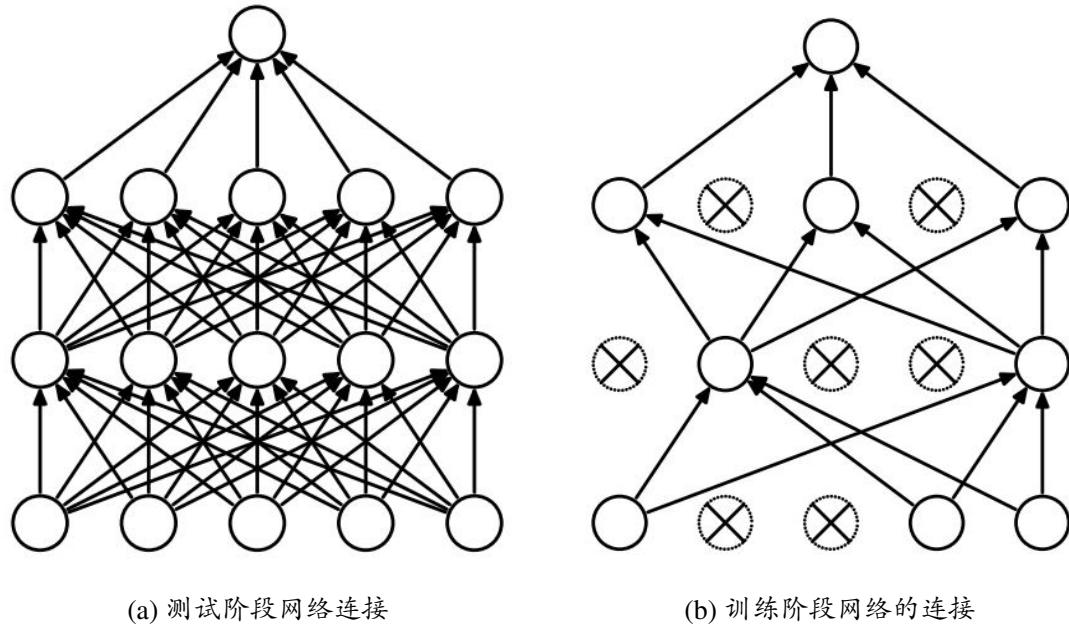


图 3-5 Dropout 神经网络模型

Figure 3-5 Dropout Neural Net Model.

差偏移指的是在网络训练的过程中，隐藏层输入的分布一直在变，导致网络收敛变慢。批量标准化的方法可以使隐藏层的输入分布稳定并近似服从标准正态分布，很多激活函数（如  $\tanh$  函数和  $\text{sigmoid}$  函数等）在 0 值附近的梯度是最大的，因此采用批量标准化的方法可以获得一个较大梯度，从而大大提高了网络的训练速度。另外，采用批量标准化的方法还可以降低对初始学习率选取的敏感度。

### 3.4 基于深度卷积网络的线虫前景轮廓分割

秀丽隐杆线虫由于通体透明，且 PDMS 制作的微流控芯片也是透明的，所以线虫轮廓的分割是个难点。通过简单的阈值分割的方法，并不能够很好的分割线虫的轮廓，因为前景像素的灰度值范围被包括在背景像素的灰度范围内。通过背景减除的方法分割的线虫轮廓。由于图像噪声的影响，会导致分割的线虫轮廓出现断裂、不完整等情况。这些都会导致线虫跟踪的失败。另一方面，背景的估计需要 50 ~ 100 帧背景不变的图像。当需要对线虫进行实时跟踪时，基于背景估计的线虫前景分割算法需要一个启动时间用于背景的估计。另外，当 CCD 相机或载物台移动时（如：需要观察不同腔室中的虫子），由于背景的改变，这种算法还是会失效。卷积网络作为一种强大的特征提取器在许多计算机视觉任务中取得了很大的成功。

本课题将卷积网络用于线虫轮廓的前景分割具有如下优势：

1. 与基于背景减除的方法相比，基于卷积网络的分割不需要对图像背景建模，只依赖当前帧的图像，从而能够保证实时性的要求。
2. 能够降低硬件成本，传统的线虫图像处理，为了获得一个背景和线虫轮廓对比度比较高的图像。通常使用特制的硬件对 CCD 和照明都有很高的要求。卷积网络作为一种强大特征提取器降低了对图像质量的要求。
3. 鲁棒性更好，传统的线虫轮廓分割算法，通常需要人工的选取一些超参数（如：分割的阈值，形态学操作中核的大小等等），但由于视频采集过程中照明的变化以及图像噪声的影响很难选取一个最佳的全局参数。基于卷积网络的分割是一种端到端的方法，输出直接是分割的结果，因此，这种方法几乎不依赖超参数的选取，具有很好的鲁棒性。

### 3.4.1 传统的图像分割方法分类介绍

图像分割是生物医疗图像处理任务最重要的步骤之一，图像分割指的是将图像按照预先定义的相似度准则将图像划分为若干个区域的过程，相似度准则可以是代表同一个对象的像素的特性或特征。

#### 3.4.1.1 基于像素的图像分割方法

基于像素的阈值分割方法通过计算图像灰度直方图的统计信息来确定一个或者多个阈值，并对图像中每个像素进行比较并分类。其中 OSTU 于 1979 年提出了基于最大类间方差的阈值分割方法<sup>[52]</sup>，通过最大化类间方差计算出一个全局最优阈值。与全局阈值法相对应的是自适应阈值的分割，由于每个像素的阈值是根据其局部邻域内的统计直方图计算的，所以自适应阈值分割方法对于光照不均匀的图像其具有很好的分割效果。除了以上提到的两种阈值分割方法外，还有最大熵阈值法、双峰法、P 参数法等阈值分割方法。

与阈值分割方法不同，基于像素聚类的分割方法是通过对像素的灰度值或者特征向量的聚类来实现图像分割的。如彩色图像往往具有 R、G、B 三个通道，因此可以将每个像素的三个颜色通道的分量值添加到该像素关联的特征向量里，在特征向量空间根据相似度的度量对像素进行聚类实现图像分割。这是像素聚类的一个优势，其能够利用每个像素更多的信息，而阈值分割只能利用像素的灰度值。像素聚类往往会产生不连续带有孔洞的区域或者具有单个孤立像素的区域，所以聚类后一般都需要后处理步骤以消除这些缺陷。常用的后处理方法有区域生长、像素连接和一些基于规则的算法等。常见的聚类算法有 K 均值聚类算法<sup>[53]</sup> 和以模糊集合为理论基础的模糊 C 均值聚类算法<sup>[54]</sup>(Fuzzy cmeans,FCM) 等。

### 3.4.1.2 基于边缘的图像分割方法

基于边界的图像分割算法通常使用空间滤波的方法计算图像的一阶梯度和二阶梯度。Sobel 算子、Prewitt 算子和 Robert 算子常被用于计算图像的一阶梯度信息。Laplacian 算子和 Kirsch 算子常被用于计算图像的二阶梯度信息。使用这些微分算子对图像进行空间滤波便可以将图像中不同区域的边缘检测出来。由于图像中噪声的影响，这些边缘通常是不连续的。为了形成闭合的区域，通常需要将这些不连续的边连接起来。常用的算法有 Canny 边缘检测算法。

### 3.4.1.3 基于区域的图像分割方法

基于区域生长的图像分割方法是根据预先定义的相似性准则将相邻的像素或相邻的区域融合形成一个大的区域。当原图像被过度分割成很多小区域时，区域生长的方法作为一种有效的后处理方法能够处理分割过程中目标轮廓出现断裂的情况。常用的区域生长方法有对称区域生长以及将模糊连接度方法和区域生长方法相结合的分割算法等。与区域生长相对应的是区域分裂，这种方法可以将一个大的区域按照像素或区域之间的相异度准则分割成两个或者更多不相交的小区域，区域分裂的分割方法可能使目标轮廓产生孔洞。在实际应用中，通常将区域分裂方法和区域生长方法结合使用，其中四叉树分解法是一种典型的区域分裂合并分割方法。

### 3.4.1.4 交互式的图像分割方法

所谓交互式的图像分割方法指的是需要用户提供输入的分割方法(如：需要用户提供一个矩形框或者鼠标点击等输入操作)，是一种半自动化的图像分割方法。目前常见的交互式图像分割方法有基于图论的 GrabCut 算法<sup>[55]</sup> 和基于曲线演化的活动轮廓分割方法<sup>[56, 57]</sup>。GrabCut 算法是由 Rother 等人于 2004 年提出，通过引入 Gibbs 能量函数，将前景背景分割问题转化为一个能量最小化的问题，通过迭代计算的方式将能量最小化。在分割的过程中，用户还可以用画刷将某个区域标记为前景或背景，然后让算法基于已经分割的结果继续迭代以改善分割的性能。基于活动轮廓的分割方法总是能够获得连续的闭合的轮廓。特别是将水平集理论应用于活动轮廓分割使得该方法可以自动处理轮廓的分裂和合并，使其更具灵活性。其原理是曲线在某些力(如：曲率力、梯度力等)的联合作用下呈现扩张或者收缩的运动，最终会收敛于目标轮廓的边缘。但是活动轮廓的分割方法需要提供一个初始轮廓作为算法的输入，因此也是一种半自动化的图像分割算法。

### 3.4.1.5 基于背景减除的分割方法

Zivkovic 等人<sup>[58]</sup>在 2006 年提出了一种基于对单个像素进行混合高斯建模的背景估计的方法。其主要针对背景固定的视频。以上小节提到的分割方法都只需要用到当前帧图像的信息，而与上面提到的分割算法不同，基于背景减除的分割方法还需要用到当前帧之前的多帧图像信息，因此是一种基于视频的轮廓分割方法。其原理是对图像中每一个像素用一个混合高斯概率函数建模。模型参数包括高斯均值  $u_i$ 、高斯方差  $\sigma_i$ 、高斯权重  $w_i$ 。每读取一帧图像，便对模型做一次更新。预测背景的时候，将所有高斯按照其权重的大小排序，取前  $B$  个高斯分量作为背景， $B$  由公式3-2表示，其中参数  $T$  表示背景所占的比例。得到每一帧图像的背景后再与当前帧图像相减，经过二值化处理后便可得到分割后的前景轮廓。

$$B = \arg \min_b \sum_{k=1}^b w_k > T \quad (3-2)$$

## 3.4.2 数据集的制作

### 3.4.2.1 数据集的采集

图3-6是线虫图像采集的系统装置图，主要由显微镜、CCD 相机、照明系统和微流控芯片四个部分组成，CCD 相机通过数据线连接电脑进行图像数据传输。线虫通过注射泵注入到微流控芯片的腔室内。移动显微镜的载物台使线虫腔室位于视野的正中央。调节显微镜的放大倍数使线虫腔室充满整个视野。调整显微镜焦距使视野中呈现清晰的图像。调整 CCD 相机的曝光时间以获得一个对比度比较高的线虫图像。完成以上步骤后便可以进行线虫视频的采集。

### 3.4.2.2 数据集的标注

高质量的数据集对网络的训练来说至关重要，且数据集的质量决定了神经网络模型性能的上限，通过优化网络架构的方法也只能逼近这个上限。但数据集的标注通常是一个非常耗时的过程，特别是图像分割任务要对不同的区域标记。目前很多的图像分割标注工具(如：Labelme<sup>1</sup>和 Ratsnake<sup>2</sup>等)都是采用多边形近似的标注方法，即在轮廓的四周边缘采集足够多的点，这些点构成的多边形为标注对象的轮廓。由于线虫形态变化复杂，相对于其他目标的标注往往需要采集更加密集的点才能满足线虫轮廓标注的精度。为了提高线虫图像标注的效率，本文采用了一种半自动的线虫轮廓标注方法。将 Grabcut 算法用于线虫轮廓的标注，只需要

<sup>1</sup><https://github.com/wkentaro/labelme>

<sup>2</sup><https://is-innovation.eu/ratsnake/>

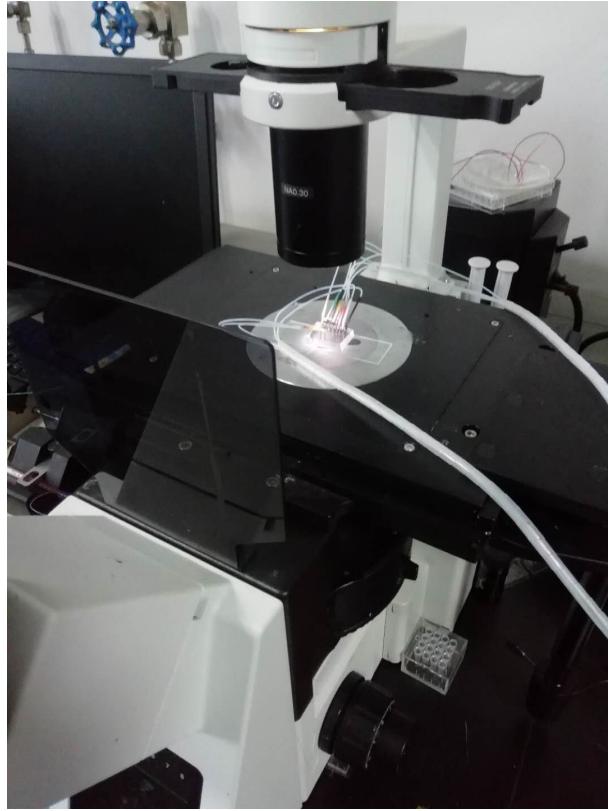


图 3-6 图像采集系统装置图

Figure 3-6 System diagram of image acquisition

用矩形框将线虫轮廓框出作为 Grabcut 算法输入，算法可以自动的分割出线虫的轮廓，最后再将多个线虫轮廓合成为一个标签图像。通过这种方法，本文制作了一个包含 236 个样本的数据集，图3-7是数据集部分示例。整个数据集按照 8 : 2 的比例将其分为训练集与测试集两部分分别用于网络模型的训练与测试。

### 3.4.3 条件随机场模型在分割任务中的应用

条件随机场 (Condition random field, CRF) 是概率无向图模型中的一种<sup>[59]</sup>，目前被很多研究者用于解决图像分割问题并取得了很大的成功<sup>[60-62]</sup>。其能够对空间中相邻像素之间的关系进行建模，从而可以得到具有空间一致性和更加精细化的分割结果。如果分割算法没有考虑到相邻像素之间的依赖关系，相当于认为空间中每个像素都是独立的，这样会导致分割结果中被分割对象的不连续以及可能丢失掉很多细节结构。本文将条件随机场模型应用于线虫前景轮廓的分割任务中，并通过卷积模块加以实现，这样可以保证整个模型的可微性。

线虫图像的前景背景分割问题本质上是一个二分类问题，假设有一个标签  $Y =$

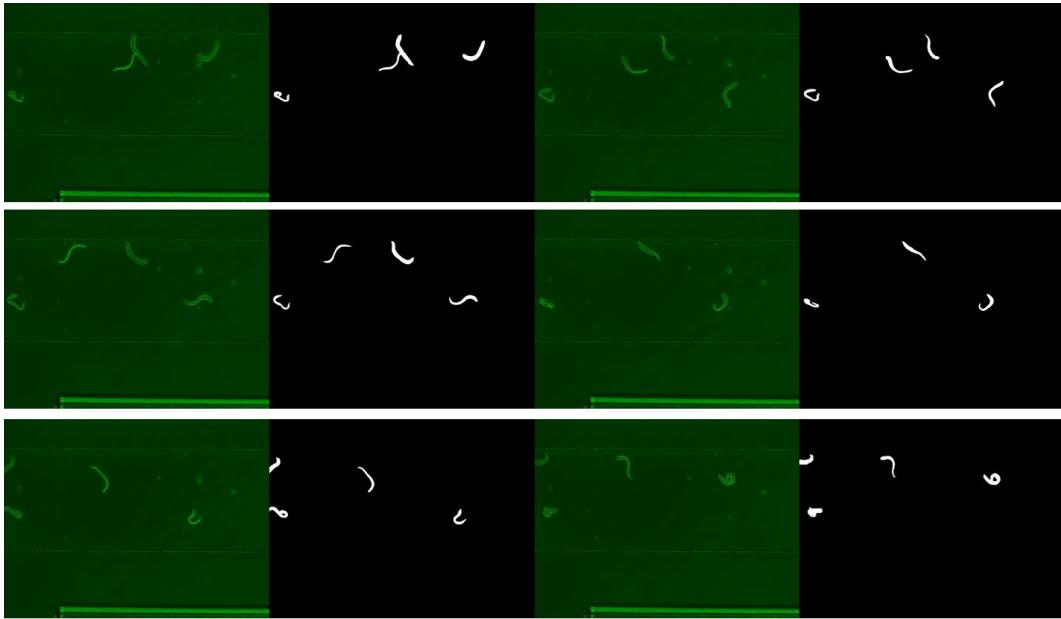


图 3-7 线虫前景轮廓分割数据集示例

Figure 3-7 The image examples from C.elegans foreground segmentation dataset

$\{Y_1, Y_2, \dots, Y_N\}$ ,  $N$  表示总的像素,  $Y_i \in \{0, 1\}$  则标签  $Y$  总共有  $2^N$  种取值。在  $Y$  的所有取值中, 概率最大对应的取值即为最优分割的结果。图像分割问题被转化为基于条件随机场的最大条件概率问题。根据吉布斯分布与条件随机场的等效性<sup>[63]</sup>, 定义如下基于标签  $Y$  的能量函数:

$$E(Y|I) = \sum_l Y_l U(l) + \sum_{l,k} Y_l W_{l,k} Y_k \quad (3-3)$$

其中一元项  $U(l) = g(h, l)$  表示第  $l$  个像素标签  $Y_l$  取值为 1 的损失,  $h$  表示图像特征,  $W_{l,k}$  表示标签  $Y_l$  和标签  $Y_k$  同时出现的权重。给定一幅线虫图像  $I$ , 对应的标签为  $Y$  的条件概率为:

$$P(Y|I) = \frac{1}{Z} \exp(-E(Y|I)) \quad (3-4)$$

其中  $Z$  为归一化项。通过平均场近似<sup>[60]</sup> 的方法, 标签  $Y_l = 1$  可以通过以下迭代的方法求出:

$$\Phi(Y_l = 1)_t = \sigma \left( U(l) + \sum_k W_{l,k} \Phi(Y_k = 1)_{t-1} \right) \quad (3-5)$$

其中  $\Phi(Y_l = 1)$  表示  $Y_l$  取值为 1 的概率,  $\sigma(a) = 1/(1 + \exp(-a))$  表示 sigmoid 函数。 $U(l)$  通过基于图像特征  $h$  的卷积获得。 $\sum_k W_{l,k} \Phi(Y_k = 1)_{t-1}$  是通过  $t - 1$  阶段的概率图  $\Phi_{t-1}$  与卷积核  $W$  的卷积操作实现的。在  $t$  时刻的概率图  $\Phi_t$  可以由如下公式计

算：

$$\Phi_t = \mathcal{M}(U, W^k) = \begin{cases} \sigma(W^k * U), & t = 0 \\ \sigma(W^k * \Phi_{t-1} + U), & t = 1, 2, 3 \end{cases} \quad (3-6)$$

其中  $\mathcal{M}$  表示基于共享权重的递归卷积， $W^k$  表示共享卷积核，其可以对相邻像素之间的空间关系进行建模。本文中，我们使用了 3 次递归卷积实现条件随机场的近似。

#### 3.4.4 网络结构的设计

本文提出了一种用于线虫前景轮廓分割的基于条件随机场的卷积网络模型，图3-8为对应的网络架构。在卷积分割网络的设计中，融合多种尺度的图像信息对分割任务来说是相当重要的（如：U-net 网络<sup>[64]</sup> 和 SegNet 网络<sup>[65]</sup> 架构的设计），充分利用不同尺度的信息能够提高目标的定位精度从而减小分割误差，本文在分割网络架构的设计上也结合了多种尺度的信息。在图3-8中，网络的前半部分通过降采样操作引入新的分支，网络的后半部分通过上采样操作与上游的分支融合。网络输入张量的尺寸为  $592 \times 800 \times 3$ （代表一幅 RGB 彩色图像），通过不断分支并降采样，网络在最底层的分支上达到最小尺度（分辨率为  $37 \times 50$ ）。每次降采样都将分辨率降低为原来的一半，同时将特征通道数扩大为原来的两倍。最终网络与最上层的分支融合得到尺寸为  $592 \times 800 \times 16$  特征图，将特征图与上一章节介绍的条件随机场模块相连接构成整个分割网络的结构。

残差网络<sup>[66]</sup> 能够在增加网络深度的同时使网络依然易于训练，更深的网络往往能够学习到更加抽象的特征从而提高网络的性能。因此本文使用了如图3-9所示的残差连接模块<sup>[67]</sup> 作为图3-8中的基本连接单元，这种残差模块包含三个连接通路。中间的通路由于使用了降采样，在卷积核尺寸保持不变的情况下，输出神经元的感受野将扩大到原来的两倍。但由于使用了降采样，所以导致分辨率下降。但上下两条路径上包含高分辨率的信息。最后将这三个路径的输出叠加在一起作为残差连接模块的输出。这种连接方式在扩大网络的感受野的同时依然保持原来的分辨率。

#### 3.4.5 网络模型的训练

在线虫的实时跟踪任务中，神经网络模型的复杂度和实时性是需要关注的重点。太复杂的网络其推断时间耗时太长往往达不到实时性的要求，因此必须要在网络结构的设计时加以考虑。而网络的训练和推断所消耗的时间的评估依赖于所采用的机器学习库和网络模型运行的硬件平台。因此为了评估不同网络模型的复

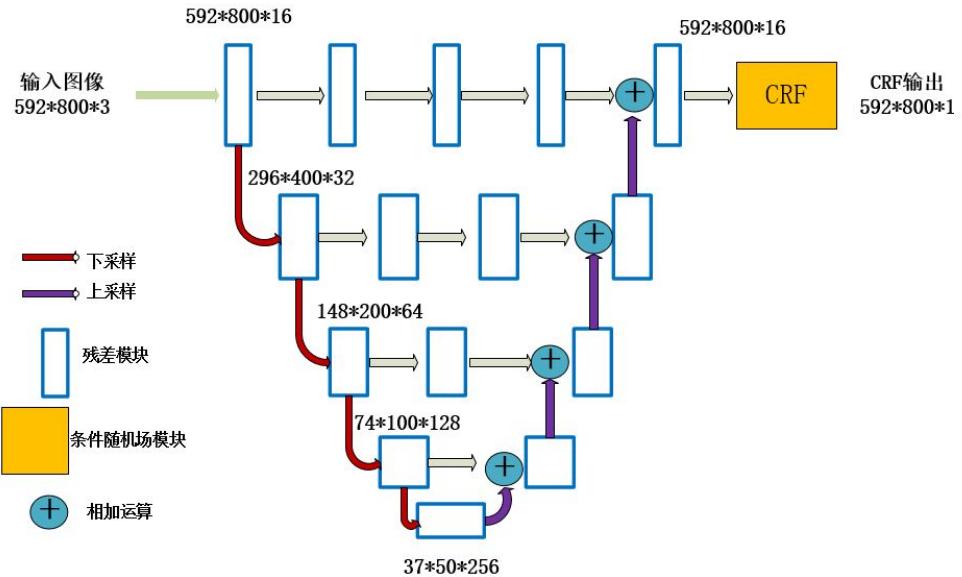


图 3-8 分割网络架构图

Figure 3-8 The Architecture of segmentation network

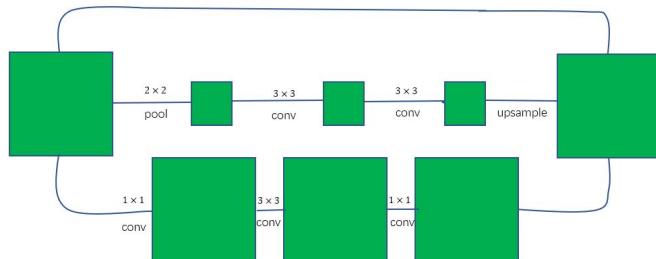


图 3-9 残差连接模块

Figure 3-9 Residual connection module

杂度和实时性，表3-1 列出了本文中所有网络模型运行的硬件平台。

增强训练数据的多样性能够有效提高网络的泛化性能，特别是在较小的数据集上网络可能会出现过拟合的情况。为了提高网络的泛化能力，在网络训练阶段本文对训练集进行了数据增强。本文首先对训练图片以一定概率随机地进行水平镜像翻转或垂直镜像翻转，并将对应的标签做相同的变换。最后将训练图片加入一定量的高斯白噪声，以提高数据的多样性。在 batchsize 参数的选择上，考虑到本文训练图片的分辨率较高（为  $592 \times 800 \times 3$ ）以及网络训练采用的硬件 GPU 显存较小（4g），因此不能将 batchsize 设置过大，本文这里将 batchsize 设为 1，即采用单样本更新的方式训练网络。假设网络的最后一层输出是一张概率图设为  $\Phi$ ， $y$  为

标签图(只包含0和1)。公式3-7表示网络训练的交叉熵损失,通过随机梯度下降的方式不断的调整网络的模型参数使交叉熵损失不断下降。最终网络收敛到局部极小值或者全局最小值,训练损失将不会再下降。图3-10表示训练过程中,训练损失和像素准确率随迭代次数的变化。图中可以看出随着迭代次数的增加,训练损失不断减小,像素准确率不断增加。网络在第1000次迭代后,网络基本已经收敛。经测试线虫前景轮廓分割网络推断单张图片的速度为16ms。

$$\text{entropy\_loss} = \sum_{i,j} y_{ij} \log \Phi_{ij} + (1 - y_{ij}) \log(1 - \Phi_{ij}) \quad (3-7)$$

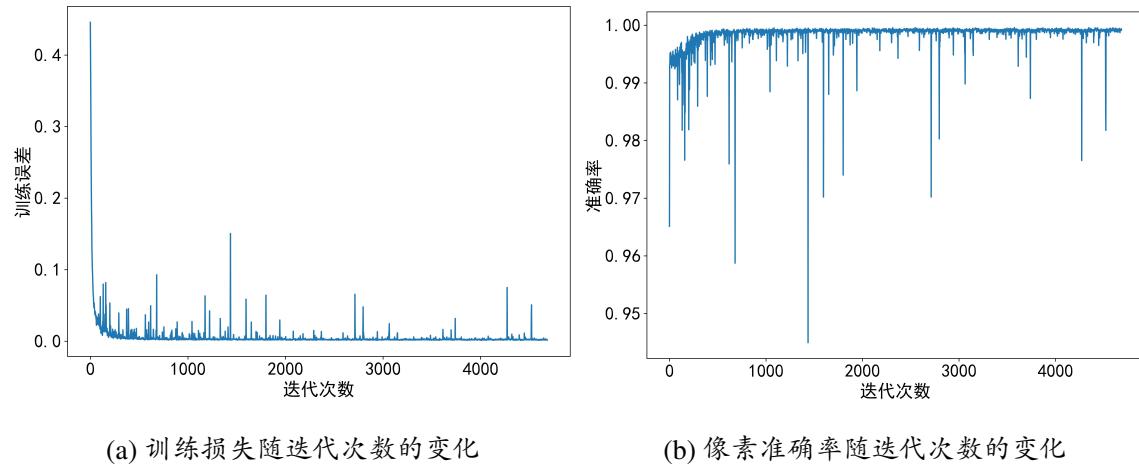


图 3-10 训练损失和像素准确率随迭代次数的变化

Figure 3-10 Training loss and pixel accuracy varied with the number of iterations

表 3-1 算法运行的实验平台

Table 3-1 Experimental platform for algorithm running

平台参数	配置
操作系统	Windows 10 家庭版
系统内存	8g
CPU	i5-6300HQ
GPU	GTX 950M
显存	4g
深度学习库	Tensorflow

### 3.4.6 实验结果分析

#### 3.4.6.1 评价指标

为了评估线虫前景轮廓分割网络的分割性能，需要一些量化指标来量化分割的结果。本文采用的三个量化指标分别为：过分割率(Over Segmentation,OR)、欠分割率(Under Segmentation,UR)以及总体分割误差率(Overall Error Rate,ER)<sup>[68]</sup>。分别由公式3–8、3–9和3–10计算。其中 $Q_p$ 表示目标像素被误分类为背景像素的数目， $U_p$ 为背景像素被误分类为目标像素的数目， $D_n$ 表示背景像素点的数目， $D_p$ 表示目标像素点的数目。

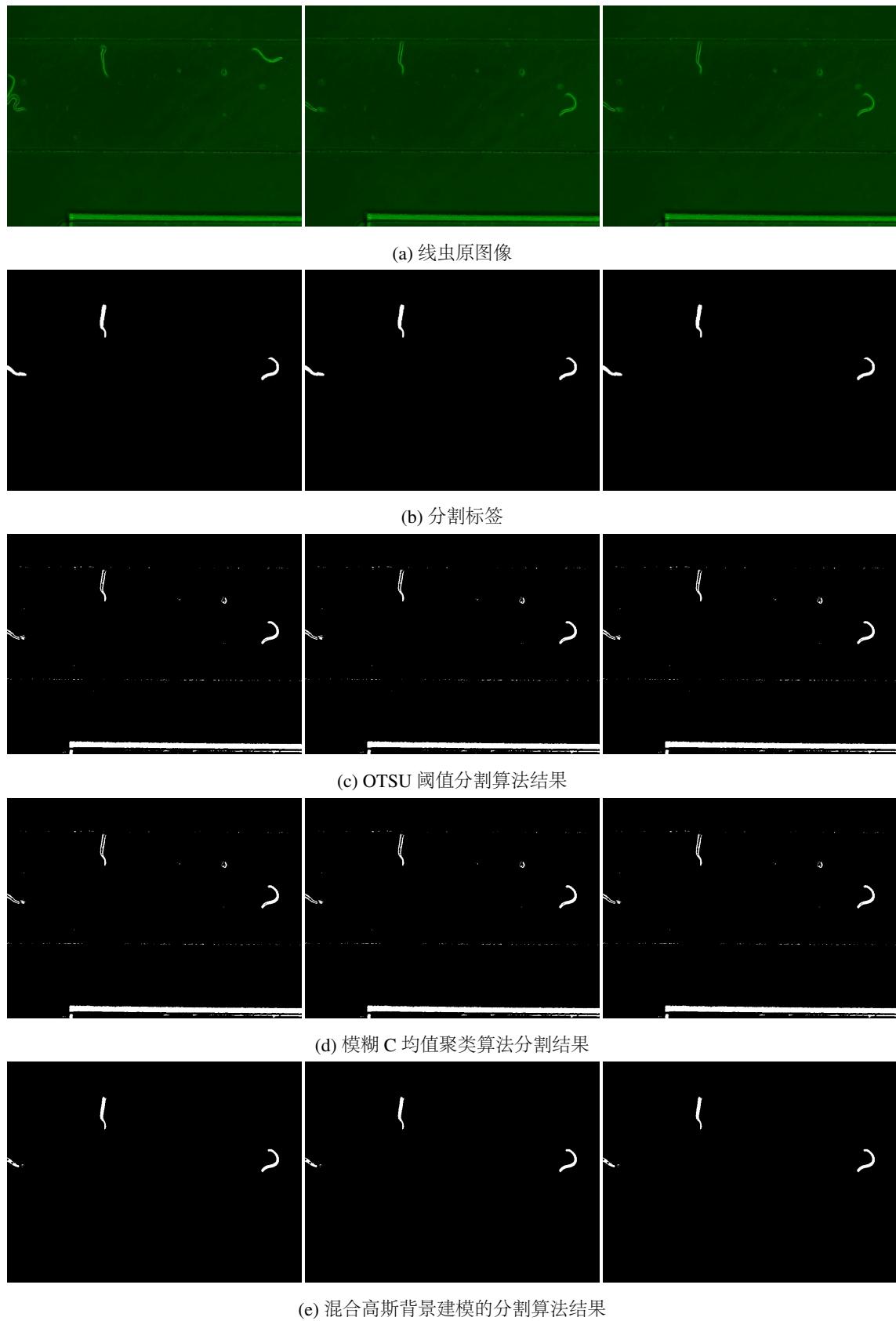
$$OR = Q_p/D_p \quad (3-8)$$

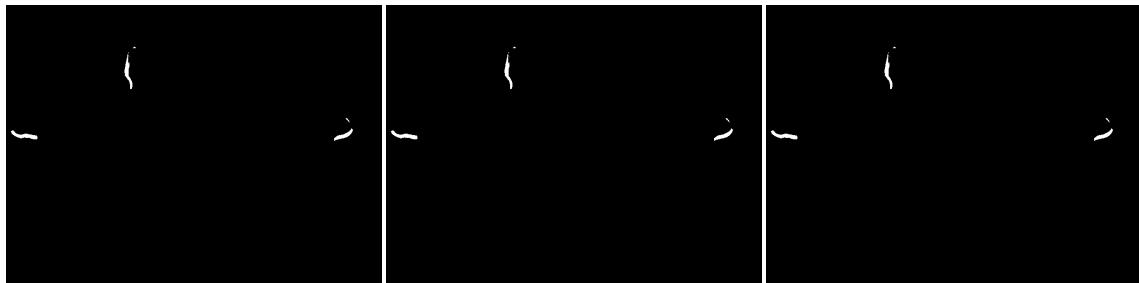
$$UR = U_p/D_n \quad (3-9)$$

$$ER = (Q_p + U_p)/(D_p + D_n) \quad (3-10)$$

#### 3.4.6.2 分割效果对比

为了比较分析不同算法的前景轮廓分割效果，本小节我们将本章提出的基于条件随机场的卷积分割算法与传统的前景轮廓提取方法进行了对比。图3–11显示了不同算法的分割结果。从图中可以看出基于模糊C均值聚类的分割方法和基于阈值分割的OTSU算法的分割结果存在大量的像素分类错误，从原图像可以看出，图像底部微流控芯片结构部分的亮度较高，但这部分区域的像素应该属于背景像素，这两种基于像素的分割方法都将这些区域分类为前景像素。另一方面，从原图可以看出线虫的轮廓中心处的亮度较低，这两种算法都将这些区域的像素分类为背景，造成了线虫轮廓的不连续以及存在孔洞。虽然经过后处理步骤可以针对单张图像过滤掉非线虫轮廓，以及运用形态学图像处理技术可以消除掉线虫轮廓中的空洞和不连续等缺陷，但在线虫视频的自动化分析任务中，由于每一帧图像的噪声情况都不一样，因此很难确定一个全局的阈值。基于高斯混合模型的背景减除方法虽然比基于阈值的分割方法和基于聚类的分割方法分割结果有很大的改善，但线虫的轮廓依然存在不连续的情况。基于背景减除的分割方法最大的不足在于其只能针对背景固定的视频，因此这种分割具有很大的局限性。图3–11(f)是本文不带条件随机场模块的卷积网络(去掉CRF模块，同时用 $1 \times 1$ 的卷积将特征的通道数变为1作为网络的输出)分割的结果。从图中可以看出去掉CRF模块后，卷积分割网络由于没有考虑到空间中相邻像素的相关性，从而导致线虫轮廓的不连续，分割性能较差。而条件随机场模块能够显著的改善线虫前景轮廓分割中的不连续等情况，与以上的方法相比，其分割性能要优于其他的方法，而且其分割的性能不依赖于超参数的选择，具有很好的鲁棒性。





(f) 本文算法分割结果 (没有条件随机场模块)



(g) 本文算法分割结果

图 3-11 不同分割算法的分割结果

Figure 3-11 The result of different segmentation methods

### 3.4.6.3 分割性能的量化分析

表 3-2 不同分割方法的性能比较

Table 3-2 Performance comparison of different segmentation methods

算法	过分割率	欠分割率	总体误差
本文算法	12.29%	0.01%	0.11%
本文算法 (无 CRF 模块)	26.93%	0.03%	0.17%
背景减除	19.55%	0.02%	0.12%
OTSU 算法	26.62%	2.12%	2.25%
FCM 算法	27.41%	2.15%	2.28%

为了定量的分析不同分割算法的性能，本文采用了三种性能指标对不同算法在测试集上分割的结果进行量化如表3-2所示。从表中可以看出基于阈值分割的 OTSU 算法和基于聚类的 FCM 算法在过分割率、欠分割率以及总体误差三个指标上均表现较差。另外，没有使用 CRF 的卷积分割网络在三个指标上的表现均弱于基于背景减除的分割方法。但在卷积网络后端加入 CRF 模块后，网络的分割性

能有了很大的提升，其分割性能要优于其他的分割算法，总体的像素误差下降到 0.11%。

### 3.5 基于深度卷积网络的线虫轮廓解析

多线虫跟踪问题中，多线虫轮廓之间相互纠缠是造成线虫跟踪丢失的主要原因，是实现单个线虫长期跟踪的关键。如图3-12所示，是一张经过线虫前景轮廓分割后得到的一幅二值化图片，图中线虫轮廓之间出现严重的相互遮挡的情况，多个单线虫轮廓融合为一个连通域。虽然人眼可以很轻松的辨别出图中所有单个线虫的轮廓，但自动化地解析出单个线虫的轮廓却十分困难，尽管深度学习在图像分割任务中取得了很大的成功，但线虫轮廓的解析不同于图像分割，因此不能直接转为一个端到端的学习问题。Yurchenko 等人<sup>[69]</sup>于 2017 年提出了一种线虫解析的算法，该算法首先在经过前景背景分割的二值化图像上随机选取许多的图像块，使其尽可能覆盖所有的线虫轮廓，并通过神经网络提取每一个图像块的特征向量。然后建立一个大的线虫姿势搜索库，库中包含大量的线虫姿势和与之对应的特征向量。将图像块对应的特征向量作为键值通过最近邻的方式在线虫姿势搜索库中搜索到最佳的匹配，从而得到线虫的姿势估计。最后通过整数规划的方式去除掉重复的线虫轮廓，从而得到线虫解析的结果。该算法的不足在于，为了使这个库能够包含各种大小和姿势的线虫，需要建立一个非常大的搜索库(大约 4 百万)。库的规模限制了轮廓解析的精度，所以只能得到近似的解析结果。另一方面在一个很大的库中搜索匹配需要大量的时间，因此实时性是该算法的另一个不足。本文提出了一种基于深度卷积网络的线虫轮廓解析算法尝试解决这一问题，与以上提到的轮廓解析算法不同，本文提出的轮廓解析算法不需要建立搜索库，而是直接将图像块输入到网络，网络可以自动输出中央线虫的轮廓。由于神经网络模型在推断阶段速度很快，本文提出的线虫轮廓解析算法具有很好的实时性能。

#### 3.5.1 线虫轮廓解析方法介绍

本文提出的线虫轮廓解析算法的关键在于设计了一个只对位于图片中央的线虫轮廓敏感的神经网络，而对非图片中央位置的线虫轮廓不敏感。当输入一幅包含多线虫轮廓的前景图像时，网络的输出为中央线虫轮廓的图像。形式化的描述如下：假设一幅图片中包含  $N + 1$  个线虫轮廓，定义一个集合  $E = \{e_0, e_1, \dots, e_N\}$ ，其中  $e_0$  表示轮廓位于图片中央的线虫， $e_i$  包含线虫重心坐标、轮廓以及方向等信息。 $I(E) = R(\{e_0, e_1, \dots, e_N\}; \xi)$  表示由集合 E 渲染得到的包含  $N + 1$  个线虫的图片，R 表示渲染函数， $\xi$  表示随机噪声。 $I(E_0) = R(\{e_0\}; \xi)$  表示由中间线虫渲染得

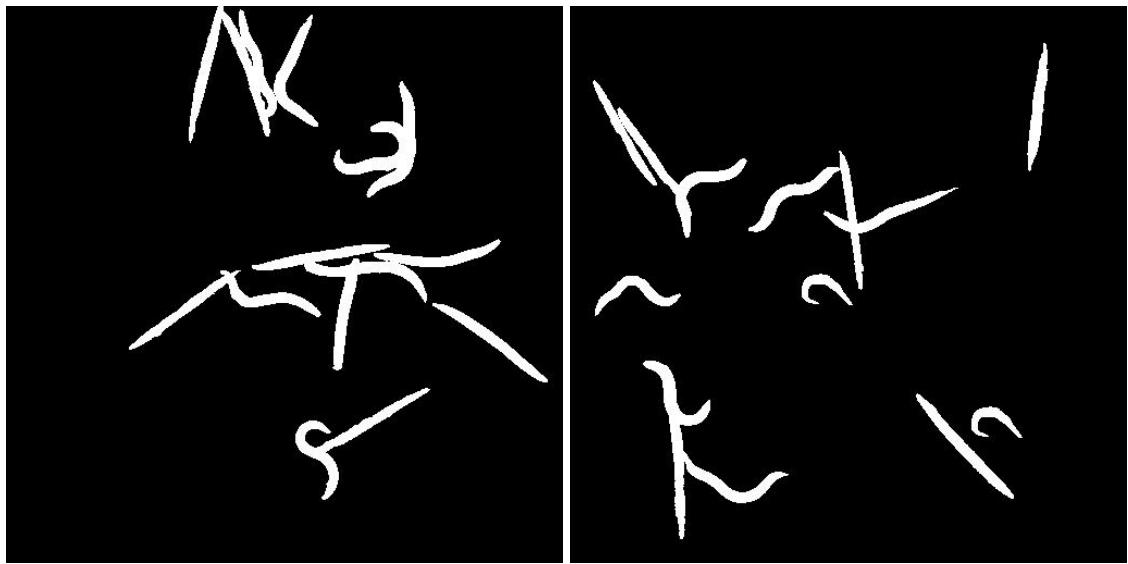


图 3-12 多线虫轮廓相互纠缠示例

Figure 3-12 Example of multi-worm entangled with each other

到的图片。我们希望得到这样的映射  $S : I(E) \rightarrow I(E_0)$ , 映射函数  $S$  通过一个深度卷积网络加以实现, 本文将其命名为 SingleOut-net 网络。

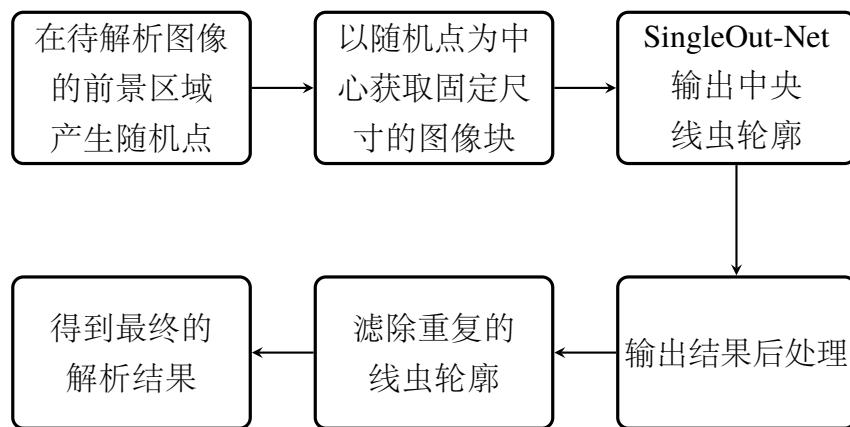


图 3-13 线虫轮廓解析的算法流程

Figure 3-13 Algorithm flow for C.elegans image parser

如图3-13是本文提出的线虫轮廓解析方法的主要流程。首先假定已经从前景轮廓分割的步骤中已经得到包含多线虫轮廓的二值化图像如图3-12所示, 轮廓解析算法的第一步是在该二值化图像的前景像素中(即非零值像素)产生足够多的随机点使得每个线虫轮廓上至少包含一个随机点。然后以这些随机像素点为中心在

原图中获取固定尺寸的图像块，将这些图像块输入到训练好的 SingleOut-Net 网络，网络输出为图像块中央位置的线虫轮廓图像。对 SingleOut-Net 输出的图像进行简单的后处理及二值化操作，然后运用轮廓提取算法即可得到这些图像块中央位置线虫的轮廓。由于在随机点产生阶段，同一个线虫轮廓包含不止一个随机点。因此由 SingleOut-Net 网络输出的线虫轮廓在原图中可能代表同一个线虫，因此需要经过轮廓滤除步骤。通过滤除掉重复的线虫轮廓以及不完整的轮廓，最终可以得到解析的结果。算法3-1描述线虫轮廓解析的整个算法实现。

---

### 算法 3-1 线虫轮廓解析算法

---

**输入:** *Worm\_Image* 待解析的线虫图像，算法的输入。

**输出:** 输出轮廓解析的结果

```

1: function PARSER_WORM(Worm_Image)
2:   Seed_points  $\leftarrow$  Generate_seed_points(Worm_Image)
3:   Image_Patches  $\leftarrow$  Crop_Image(Worm_Image, Seed_points)
4:   SingleOut_OutputImages  $\leftarrow$  []
5:   for i = 0  $\rightarrow$  Image_Patches.length - 1 do
6:     SingleOut_OutputImages[i]  $\leftarrow$  SingleOut_Net(Image_Patches[i])
7:     SingleOut_OutputImages[i]  $\leftarrow$  Post_ProcessImage(SingleOut_OutputImages[i])
8:   end for
9:   Worm_Contours  $\leftarrow$  Extracte_WormContour(SingleOut_OutputImages)
10:  resulte  $\leftarrow$  Filter_WormContour(Worm_Contours)
11:  return resulte
12: end function

```

---

### 3.5.2 随机点的产生

随机点的产生直接影响线虫轮廓解析方法的效率，随机点的数量过大导致 SingleOut-Net 网络需要对大量的图像块处理，需要很大的计算量，从而导致解析一帧图像需要大量的时间。但随机点的数量太少，可能导致有些线虫的轮廓上没有随机点覆盖，这些线虫的轮廓将得不到解析。实验发现，直接在原图像中生成随机点的方式非常的低效，主要是因为前景像素只占原图中总像素的一小部分，大部分的随机点落在了背景里。本文提出一种高效的随机点产生方法。首先，提取原图中所有的轮廓，然后在每个轮廓的边缘上等距的采样一定数量的边界点，最后在边界点的邻域内再采样。这种随机点产生的方法只需要产生少量的随机点即可覆盖所有的线虫轮廓，最终采样的结果如图3-14所示。



图 3-14 随机点采样的示例

Figure 3-14 Example of random point sampling

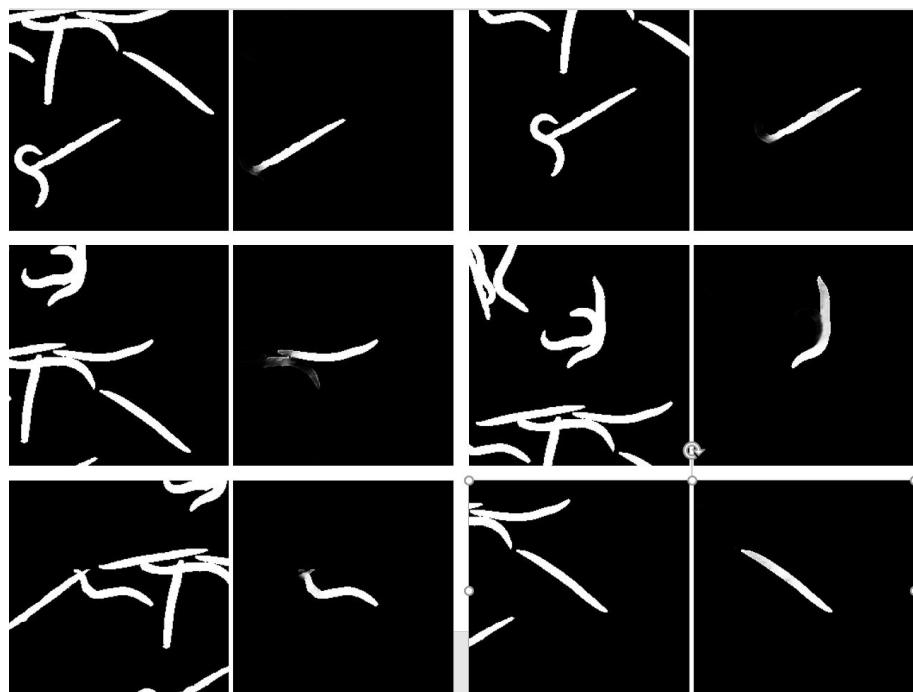


图 3-15 SingleOut-Net 网络的输入输出示例

Figure 3-15 Example of input and output of SingleOut-Net

### 3.5.3 SingleOut-Net 网络输出后处理

图3-15显示了利用训练好的 SingleOut-Net 网络对部分图像块处理的结果。从图中可以看出 SingleOut-Net 网络成功地将位于图片中央的线虫轮廓从周围的线虫轮廓中分离出来。经过二值化等后处理步骤后，再用轮廓提取算法即可得到所有图像块对应的中央线虫的轮廓。但由于在随机点生成的过程中同一个线虫轮廓上可能包含多个随机点。所以在 SingleOut-Net 输出的结果中，同一个线虫的轮廓可能出现了多次。通过计算两个线虫轮廓在原图坐标中面积的重合度可以判定这两个轮廓是否表示同一个线虫，当两个轮廓的面积重合度大于一个设定的阈值时，则只需保留其中的一个轮廓。当一个轮廓完全被另一个轮廓包含时，则保留轮廓面积较大的轮廓。过滤掉重复的轮廓以及不完整的轮廓后即可得到解析的结果，最终的解析结果如图3-16所示。

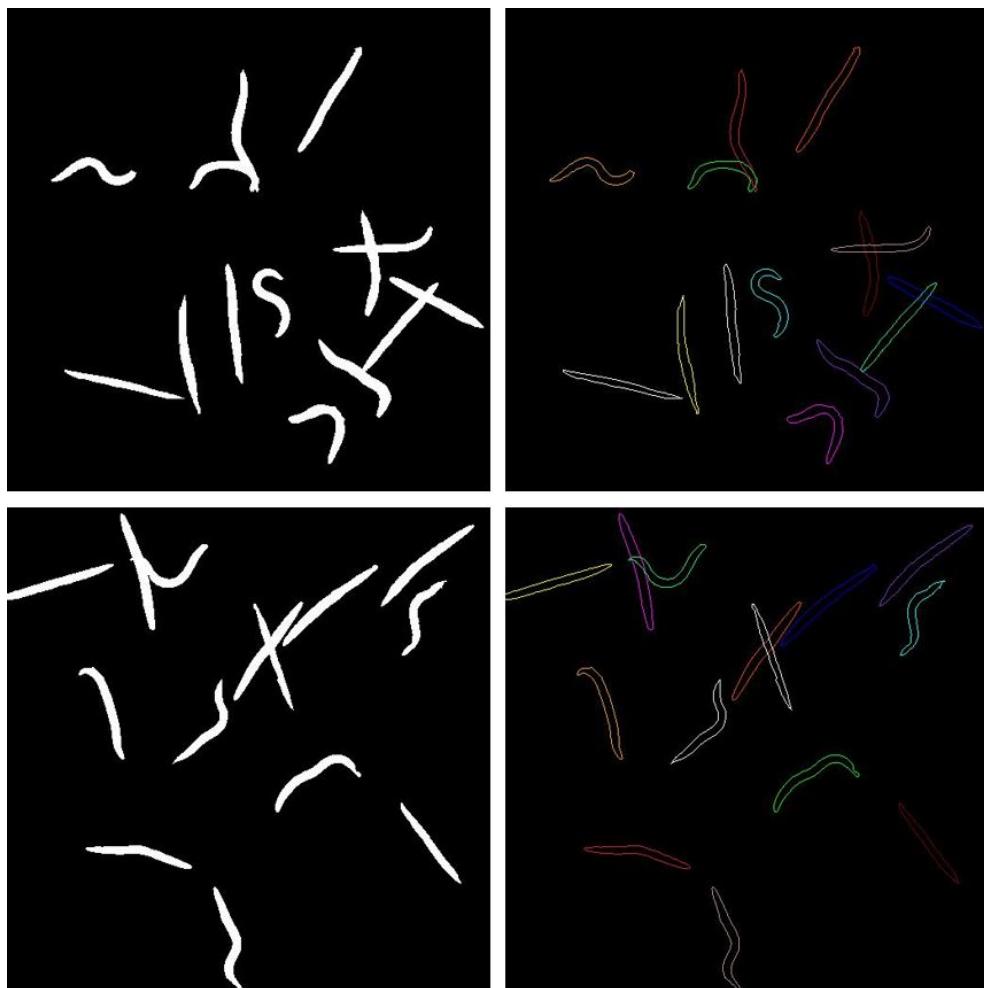


图 3-16 多线虫轮廓解析结果

Figure 3-16 Parser result of *C.elegans* contour

### 3.5.4 数据集的制作

SingleOut-Net 网络的训练需要大量标定的数据集，本文采用了人工生成的数据集来训练网络的模型参数。数据集的制作流程如图3-17所示。BBBC010 线虫数据集<sup>[70]</sup>由马萨诸塞州综合医院 Fred Ausubel 教授的实验室采集并发布，该数据集中包含 1407 张经过前景背景分割的单线虫二值化图像。数据集制作的第一步需要从 BBBC010 数据集中提取所有单线虫的轮廓得到一个线虫的轮廓库(库中包含 1407 个不同姿势和大小的单线虫轮廓)。然后从单线虫的轮廓库中随机地选取一个线虫轮廓在一个分辨率为  $256 \times 256$  的背景图(像素值全为零)的中央画出。这一步得到的图作为网络训练的标签。继续从线虫轮廓库中随机选取若干个线虫的轮廓并在标签图像中随机的画出。至此，便得到了 SingleOut-Net 网络训练的输入图像以及对应的标签。为了加快网络的收敛速度，本文将数据集的输入以及对应的标签图像归一化到 0 ~ 1 范围内，并加入一定量的高斯白噪声。通常数据集的多样性可以使神经网络学习到更多的模式，从而使网络具有更好的泛化能力。为了进一步增强数据集的多样性，本文对每个随机选取的单线虫轮廓进行随机的缩放和随机的旋转操作。

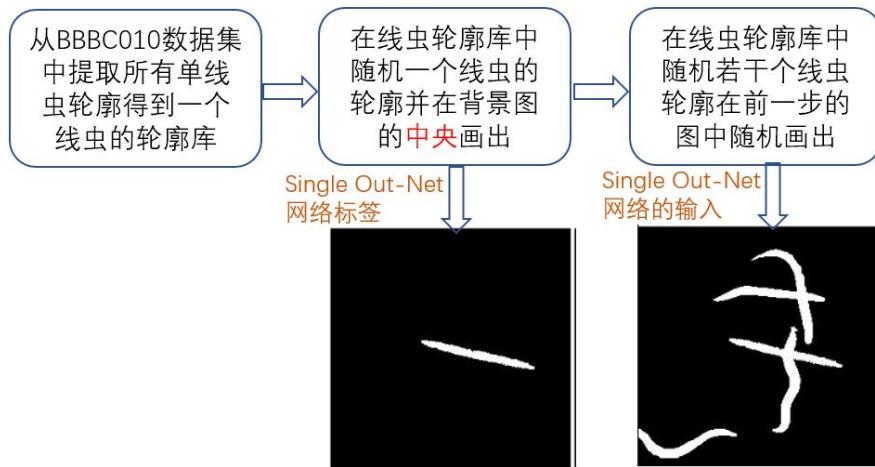


图 3-17 SingleOut-Net 网络训练数据集的制作流程

Figure 3-17 The production process of training dataset for singleout-net network

### 3.5.5 网络结构的设计

如图3-18所示是 SingleOut-Net 的网络结构。网络的输入为  $256 \times 256 \times 1$  的张量，输出也为相同尺寸的张量，整个网络架构由两个模块级联构成。第一个模块为一个类似于 U-Net 网络<sup>[64]</sup>的卷积模块，在整个网络架构中相当于一个编码器。图

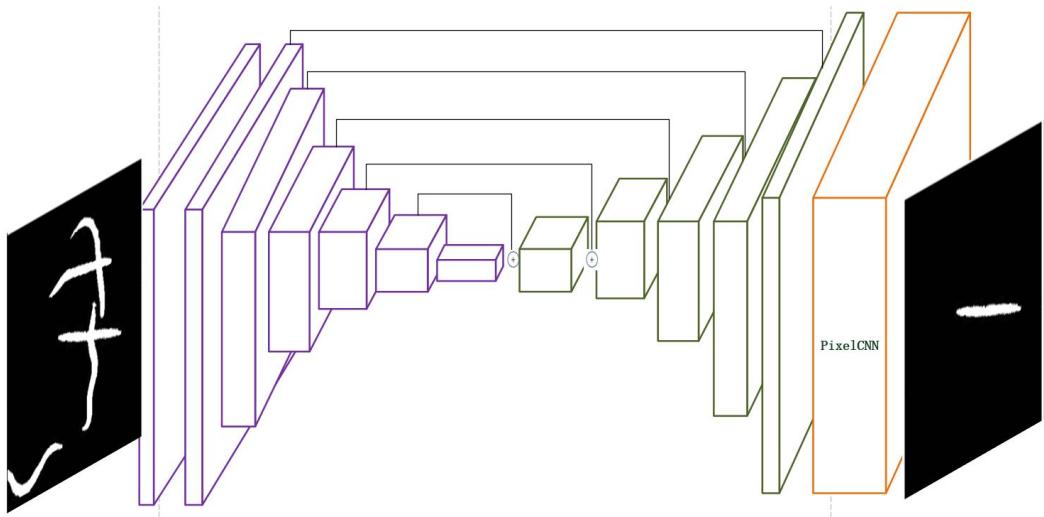


图 3-18 SingleOut-Net 网络架构

Figure 3-18 Architecture of SingleOut-Net

中的方块都表示本章3.4.4节介绍的残差连接单元。第一个卷积模块包含两条路径，分别为降采样路径和上采样路径。降采样路径上每经过一个残差连接模块后面都连接一个降采样层，降采样层将输入张量的尺寸减小一倍同时将通道数扩大一倍。降采样层通过一个卷积层实现，其卷积核的大小为 $2 \times 2$ 步长为2，卷积后紧跟着的是Batch Normalization层和激活函数层。由于Relu函数<sup>[71]</sup>具有克服梯度消失和加快网络收敛等优势，这里使用了Relu激活函数。上采样路径与降采样路径类似，只不过将降采样层换成上采样层。上采样层通过反卷积实现，其卷积核大小为 $2 \times 2$ 步长为2。在图3-18中，上采样层的输入由两部分构成，分别为降采样路径中相同分辨率的张量和上采样路径中前面的张量，图中的加号表示张量的相加操作。第二个卷积模块为PixelCNN<sup>[72]</sup>模块，将其作为解码器。PixelCNN网络由Deepmind于2016年提出并用于条件图像生成，可以生成非常逼真的图像。本文将其应用在SingleOut-Net网络中作为解码器来生成中央线虫的图像。最后通过一个 $1 \times 1$ 的卷积将特征通道数变为输出图像的通道数，并通过Sigmoid激活函数将输出的数值限制在0~1范围内。最终的输出是一个通道数为1的概率图。概率图中每一个像素值的大小表示该像素属于中央线虫轮廓的概率。

### 3.5.6 评价指标

#### 3.5.6.1 评价指标

为了更好的对不同的网络架构进行量化分析和性能比较，因此需要选取一些评价指标。本文提出的SingleOut-Net网络用于判别图像中每个像素是否属于中央

线虫轮廓像素，所以是一个像素二分类网络。本文将像素分类误差作为网络评价指标，像素分类误差由公式3-11表示。

$$\text{像素误差} = 1 - \frac{\text{被正确分类的像素数目}}{\text{总的像素数目}} \quad (3-11)$$

### 3.5.7 网络的训练

根据3.5.4节介绍的数据集生成方法，可以生成任意大小的训练集。但为了节省内存开销，本文采用了训练集动态生成的方法，即在训练阶段每个 minibatch 的样本图片都是动态生成的。但动态生成训练样本需要一定的时间开销，从而使网络训练时间变长。为了缩短网络训练时间以及在限定时间内探索更优的网络架构，本文将数据生成和网络训练这两个任务并行，即用一个专门的线程负责数据集的生成。另外为了比较不同网络架构的性能，测试集的样本应该保持不变，而本文中的数据集是动态随机生成的。为了获得一个不变的测试集，在网络训练和模型测试阶段，本文分别采用了两个不同的随机数种子初始化随机数生成器。神经网络模型的训练分为前向传播、反向传播和权值更新三个步骤。`batchsize` 作为一个重要的超参数，其决定将多少样本作为一个整体估计梯度下降的方向。如果将其设置得过大，会导致网络模型很难跳出局部最小值点；如果设置得过小，则很难获得一个准确的梯度下降方向，在本文中 `batchsize` 的大小为 4。神经网络的优化算法大致可以分为三大类：基于一阶微分的最优化方法（随机梯度下降）、基于二阶微分

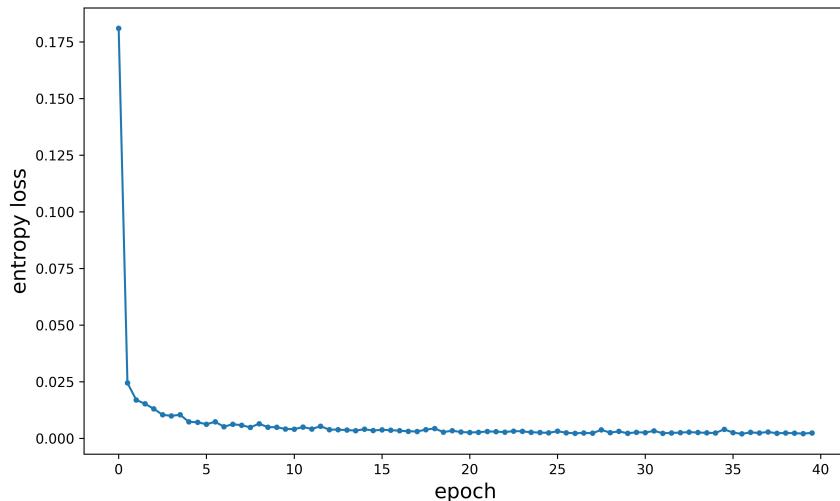


图 3-19 网络训练过程中训练损失随 epoch 数的变化

Figure 3-19 The change of training loss with epoch number in the process of network training

的最优化方法(牛顿法)以及基于二阶微分近似的方法(AdaDelta 算法<sup>[73]</sup>和 Adam 算法<sup>[74]</sup>等)。随机梯度下降的最优化方法计算量最小,但网络的收敛速度慢。牛顿法由于利用了二阶梯度信息,与其他的最优化方法相比具有更好的收敛性能,但由于要计算 Hessian 矩阵所以计算量很大。于是研究者们提出了很多基于二阶微分的近似方法,这些最优化方法是计算量和收敛性能的一个折中。本文采用了 Adam 最优化方法优化网络模型并将学习率设置为 0.0002,并将交叉熵损失作为训练的损失函数。训练 40 个 epoch(每个 epoch 包含 1000 个 batch)后网络已经完全收敛,图3-19表示损失函数随 epoch 数的增加而下降,经过 20 个 epoch 后网络已经基本收敛。

为了观察网络训练过程中 SingleOut-Net 网络性能的变化,本文对网络模型每隔 100 次迭代进行一次测试,图3-20 显示了网络训练过程中网络模型测试的结果。从图中可以看出随着网络模型迭代次数的增加 SingleOut-Net 网络逐渐学会过滤掉非中央线虫的轮廓,只保留中央线虫的轮廓。

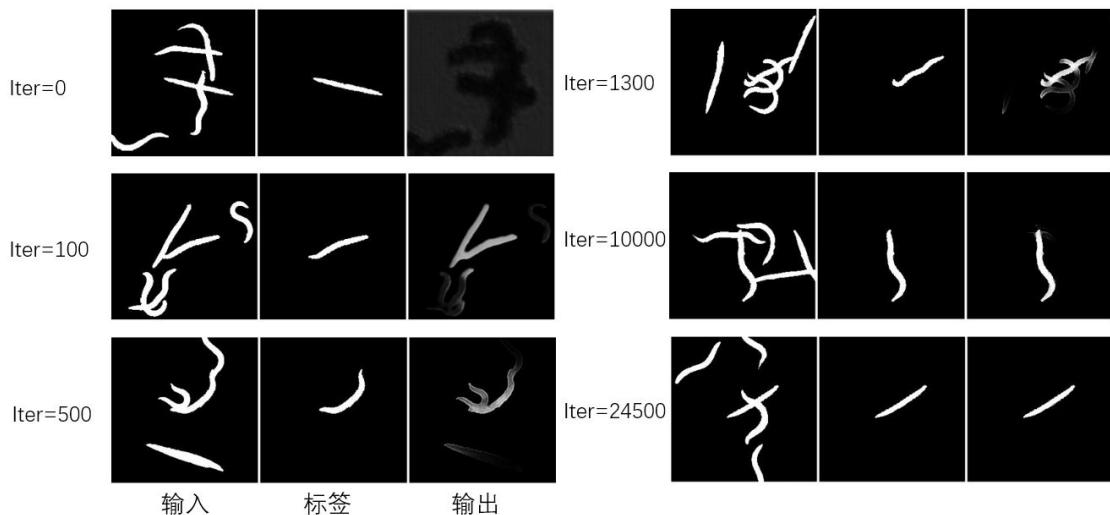


图 3-20 网络训练过程中模型测试结果

Figure 3-20 Change in contour curvature

### 3.5.8 测试结果与分析

在3.5.5节,本文介绍了 SingleOut-Net 的网络架构,由两个卷积模块构成编解码器结构。为了便于区分,本文将其命名为模型一。为了分析这种编解码结构的网络性能,本文分别从像素误差、模型复杂度和推断时间三个方面考察了 pixelCNN 卷积模块对整个网络架构的影响。由于 pixelCNN 网络模块的输入和输出的尺寸保持不变,所以将 pixelCNN 模块移去后,用一个  $1 \times 1$  的卷积将编码器模块输出的

通道数变成网络最终输出的通道数。将这种没有 pixelCNN 模块的网络架构命名为模型二。两个模型的性能比较如表3-3所示。从表中可以看出 pixelCNN 网络模块可以显著的降低像素误差(10 倍下降)，但同时由于增加了网络的深度，使得网络在推断单张图片所消耗的时间变长。

表 3-3 网络模型性能的比较

Table 3-3 A Table

网络模型	像素误差	模型大小	推断时间
模型一	0.00521%	46.8Mb	31ms
模型二	0.00045%	48.9Mb	100ms

### 3.6 线虫轮廓的跟踪

由于秀丽隐杆线虫通体透明，跟踪起来比较困难，本文采用了一种简单有效的跟踪策略。首先经过线虫前景轮廓分割和线虫轮廓解析等步骤后，可以得到每一帧图像里所有线虫的轮廓。由公式3-12和公式3-13可以计算出轮廓的重心坐标。

$$m_{ji} = \sum_{x,y} I_{x,y} x^i y^j \quad (3-12)$$

$$\vec{x} = \frac{m_{10}}{m_{00}}, \quad \vec{y} = \frac{m_{01}}{m_{00}} \quad (3-13)$$

假设当前帧有  $n$  个线虫轮廓，上一帧图像有  $m$  个线虫轮廓，由每个轮廓的重心坐标可以计算出相邻两帧图像线虫轮廓重心之间的距离，从而得到一个  $n \times m$  的距离矩阵用公式3-14 表示。

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{bmatrix} \quad (3-14)$$

矩阵中  $d_{ij}$  表示当前帧图像中的第  $i$  个轮廓的重心到上一帧图像中第  $j$  个轮廓的重心之间的距离。通过公式3-15可以得到相邻两帧图像中线虫轮廓之间的对应关系。即如果相邻两帧图像中两个轮廓重心之间的距离最短，则可以认为是同一个线虫的轮廓。在线虫跟踪过程中，每个线虫用一个 TrackID 来标识，不同的 TrackID 代表不同的线虫。

$$index(i) = \arg \min_j d_{ij} \quad (3-15)$$



图 3-21 跟踪的结果

Figure 3-21 The tracking result

但事实上由于线虫轮廓分割的不完整以及图像噪声的影响，这一策略往往失。因此，为了提高线虫轮廓跟踪的鲁棒性，在轮廓匹配的过程中本文增加了两个约束条件：

- 相邻两帧图像中同一只线虫的轮廓面积的相对变化应该小于一个阈值。
- 根据线虫运动的最大速度，同一只线虫在相邻两帧图像中轮廓重心的移动应该小于一个阈值。

当这两个条件之一不满足时，则认为跟踪丢失，此时应该分配一个新的 trackID 给当前的轮廓。算法3-3 是描述了线虫跟踪算法的实现思路。图3-21表示线虫跟踪的结果。

---

### 算法 3-2 跟踪初始化程序

---

**输入:** *Worm\_data* 双重列表，*Worm\_data[i][j]* 表示第 *i* 帧图像中第 *j* 只线虫。

**输出:** 输出 *trackID*

```

1: function INITIATE_TRACKING(Worm_data)
2:   FirstFrame_WormData  $\leftarrow$  Worm_data[0]
3:   for i = 0  $\rightarrow$  FirstFrame_WormData.length - 1 do
4:     cur_worm  $\leftarrow$  FirstFrame_WormData[i]
5:     cur_worm.trackID  $\leftarrow$  GetNewTrackID()
6:   end for
7: end function

```

---

---

**算法 3-3 线虫跟踪程序**

---

输入:  $Worm\_data$  双重列表,  $Worm\_data[i][j]$  表示第  $i$  帧图像中第  $j$  只线虫。

输出: 输出  $trackID$

```

1: function WORM_TRACKING( $Worm\_data$ )
2:   Initiate_tracking( $Worm\_data$ )
3:   for  $frame\_index = 1 \rightarrow Worm\_Data.length - 1$  do
4:      $PreFrame\_WormData \leftarrow Worm\_Data[frame\_index - 1]$ 
5:     for  $worm\_index = 0 \rightarrow Worm\_Data[frame\_index].length - 1$  do
6:        $cur\_worm \leftarrow Worm\_Data[frame\_index][worm\_index]$ 
7:        $dist\_array \leftarrow Compute\_distance(cur\_worm, PreFrame\_WormData)$ 
8:        $min\_index \leftarrow Get\_min\_index(dist\_array)$ 
9:        $Nearest\_worm \leftarrow PreFrame\_WormData[min\_index]$ 
10:      if  $\frac{|Nearest\_worm.Area - cur\_worm.Area|}{Nearest\_worm.Area} < \delta$  and  $dist\_array[min\_index] < \sigma$ 
then
11:         $cur\_worm.trackID \leftarrow Nearest\_worm.trackID$ 
12:      else
13:         $cur\_worm.trackID \leftarrow GetNewTrackID()$ 
14:      end if
15:    end for
16:  end for
17: end function

```

---

### 3.7 线虫的特征提取

线虫从头部到尾部两边近似等距的分布着 23-24 块肌肉，其头部和尾部各占其总长度的 1/6。因此线虫身体的自由度为 24。当用轮廓来描述线虫的形态时，在其轮廓四周采样 49 个点足以描述线虫所有形态。当对线虫进行特征计算时(如：计算线虫摆动频率和运动速度等)，通常是利用线虫轮廓中间的脊线进行计算。因此需要提取线虫轮廓的中线然后采样 24 个点用于特征计算。下面将首先对线虫轮廓中间脊线提取算法进行介绍，然后介绍线虫摆动频率的计算。

#### 3.7.1 线虫轮廓中间脊线提取

在得到线虫的轮廓后，将轮廓上的坐标按顺时针排列即可得到一个坐标点的循环列表。将轮廓周长的 1/48 作为一个单位边，对于线虫轮廓边缘的任意一点而

言，在其两边都可以找到一个距离为单位边长度的相邻点，这三点所成角的补角与该点的曲率成正比，因此可以用于近似曲率的计算。由于其头部和尾部的曲率往往比身体的其他部分要尖锐，所以如果将像素索引作为横坐标，曲率作为纵坐标，则这条曲线上将会出现两个波峰如图3-22所示，分别对应线虫的头部和尾部。由此便可定位到线虫的头部和尾部，另外线虫的头部曲率一般小于尾部的曲率，两个波峰中比较低的波峰对应的横坐标为线虫头部的坐标，另一个波峰对应线虫尾部的坐标。线虫头部和尾部将线虫轮廓分为两边。在其中一条边上找到所有距离另一条边最近的对应点。两条边上两对应点的中点构成线虫的中间脊线，线虫轮廓中间脊线的长度定义为线虫的体长。

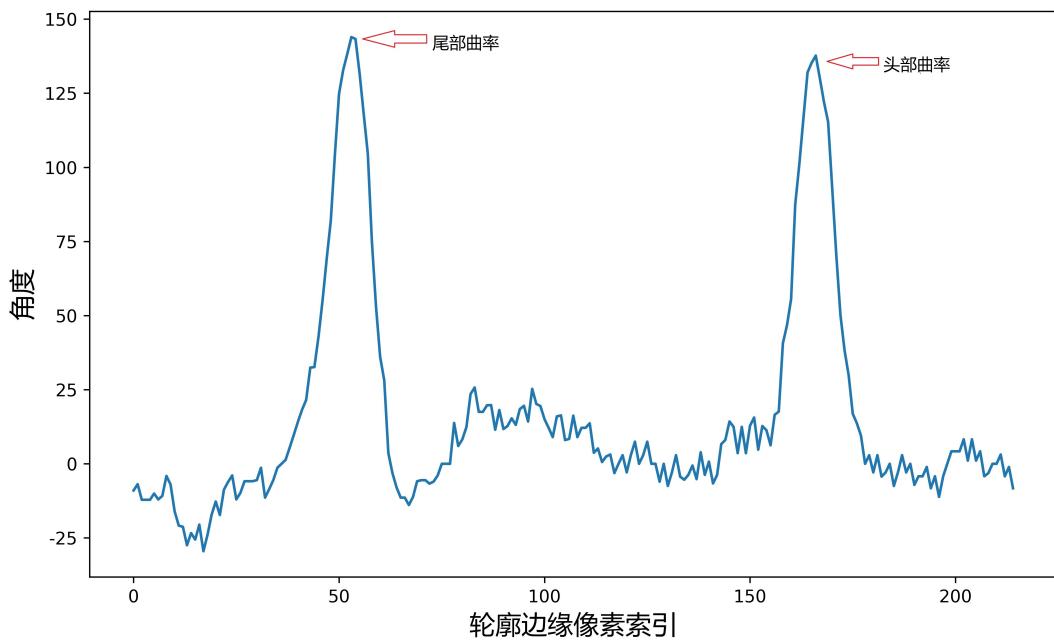


图 3-22 轮廓曲率的变化

Figure 3-22 Change in contour curvature

### 3.7.2 身体弯曲角度的计算以及摆动频率的估计

在很多毒理实验中，线虫的摆动频率经常作为一个重要的生理指标用于表征线虫的活跃程度<sup>[75]</sup>。为了计算线虫的摆动频率，我们定义一个衡量身体弯曲程度的夹角，由线虫头部、尾部和轮廓脊线的中点三点所成角定义为身体弯曲角。线虫在爬行和游动的过程中，身体弯曲角会在  $180^{\circ}$  左右振荡如图3-23，振荡的频率定义为线虫摆动的频率。在时刻  $t_0$ ，对区间  $(t_0 - \Delta t, t_0 + \Delta t)$  中弯曲角信号做 FFT 变换，假设其幅度最大值对应的横坐标为  $n$ ，则线虫在  $t$  时刻的瞬时摆动频率由公

式3-16得出。

$$\text{Swing frequency} = \frac{\text{frame\_rate} * n}{2 * \Delta t} \quad (3-16)$$

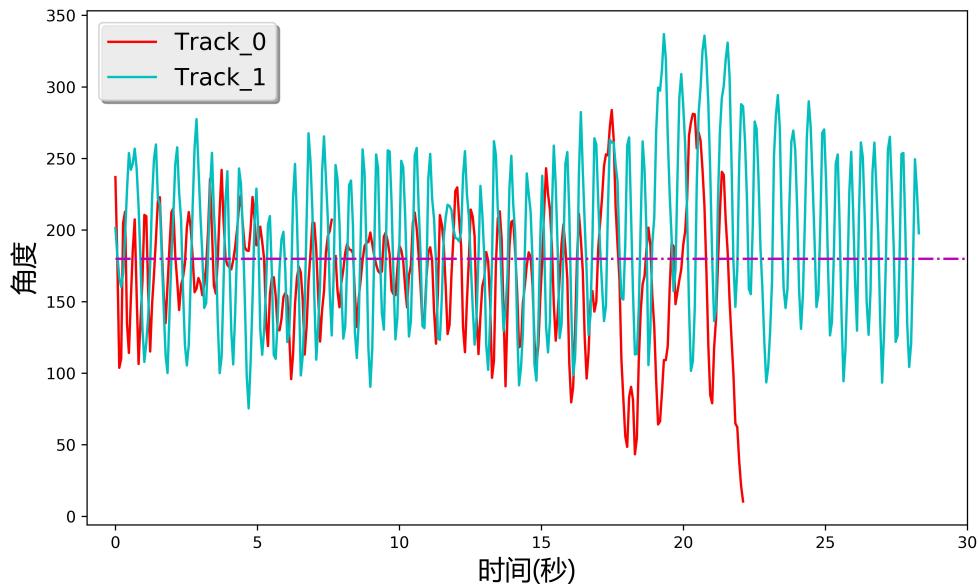


图 3-23 线虫弯曲角度的变化

Figure 3-23 Change in the bending angle of C.elegans

### 3.8 本章小结

## 第四章 线虫的特征提取和急性氧化应激实验

### 4.1 引言

### 4.2 线虫的氧化急性应激实验

氧化应激 (Oxidative Stress, OS) 指生物体氧化与抗氧化作用的失衡，当生物体被内外环境中存在有害化合物刺激时，其体内所产生的活性氮自由基和活性氧自由基将会导致细胞或者组织发生生理和病理反应。过氧化氢 ( $H_2O_2$ ) 溶液作为一种强氧化剂经常被用于线虫的氧化应激实验中。本文将野生型 N2 秀丽隐杆线虫的 L1 期幼虫作为研究对象，通过本文前面介绍的软硬件平台，研究不同线性浓度梯度的双氧水溶液对 L1 期幼虫活性的影响。

#### 4.2.1 线虫的同步化

为了得到处于同一发育阶段的幼虫需要对线虫进行同步化处理，首先用经过高压灭菌的 M9 缓冲液将 NGM 平板上混合发育期的线虫冲洗到 1.5ml 的离心管中，离心后去掉上清液，加入碱裂解液 (体积比为 1: 2 的 5N NaOH 溶液和 5%NaClO 溶液，现配)，当线虫全被腐蚀时，液体将变得清澈。再经过离心处理，去掉上层碱裂解液加入 M9 缓冲液，离心洗涤 1 到 2 次。去掉上层缓冲液并用吸管将线虫卵接种到 NGM 平板上，至此便完成了同步化操作，等线虫卵孵化便得到同步化的个体。

#### 4.2.2 线性梯度稀释芯片的操作

图4-1是实验硬件部分连接示意图，芯片上所有的阀门控制和进样控制均由 Arduino 单片机通过 uln2803 集成芯片控制多路电磁阀实现。将含有 L1 期线虫的溶液离心去上清得到线虫浓缩液，然后用移液枪加入 0.25% 的琼脂糖溶液作为线虫助悬剂。打开 6 号阀门，采用压力进样的方式将线虫溶液从 4 号进样口打入第三列腔室。打开 4 号阀门用压力进样的方式将水从 3 号进样口打入第二列腔室。然后打开 2 号阀门用压力进样的方式将 30mM 的过氧化氢溶液从 2 号进样口打入第一列腔室。然后关闭 2 号、4 号和 6 号阀门，打开 1 号、3 号、5 号和 7 号阀门，并在一号进样口施加一个周期性的气压。通过振荡的方式使前三列腔室中的液体充分混合。最后通过进样口 1 将混合好的液体打入第四列腔室，根据芯片腔室的尺寸设计，可以计算出混合后各腔室的过氧化氢溶液的浓度从上至小依次为：18mM、

16mM、14mM、12mM、10mM、8mM、6mM、4mM、2mM。并用 CCD 相机每隔 10 分钟采集线虫在 9 个腔室中的运动视频。

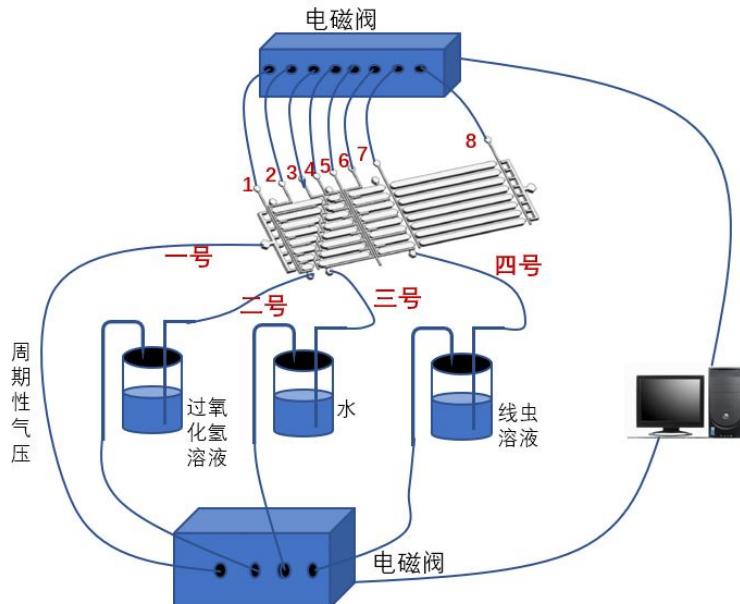


图 4-1 实验系统装置示意图

Figure 4-1 Change in contour curvature

#### 4.2.3 秀丽隐杆线虫急性氧化应激实验结果

通过本文提出的线虫轮廓分割、解析、跟踪及特征提取算法对每个腔室的各个时刻的视频进行分析并提取摆动频率特征，图4-2为了各腔室中线虫的平均摆动频率随时间的变化，可以看出随着过氧化氢溶液浓度的升高，线虫的摆动频率下降，且在同一浓度下，线虫的摆动频率随着时间而下降。为验证芯片实验的准确性，我们也在 96 孔板上手动稀释形成上述浓度后，通过对各个时间点线虫平均摆动频率的统计，实验结果表明 96 孔板实验和微流控芯片结果一致。但是相比 96 孔板实验，我们的微流控芯片平台在试剂消耗、自动分析等上面都体现了较大的优势。

### 4.3 本章小结

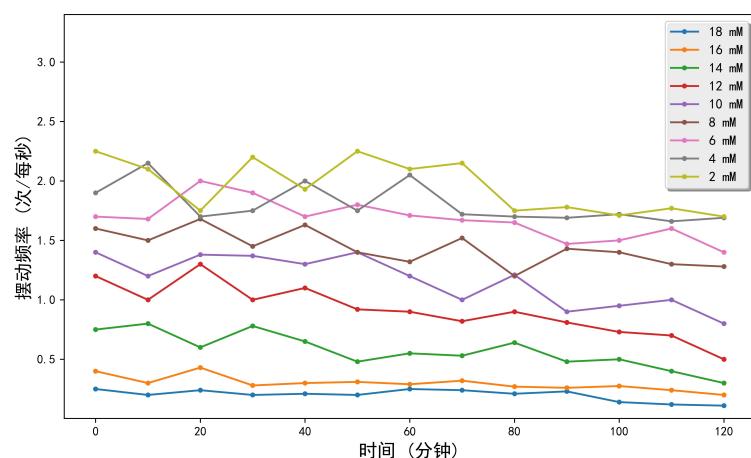


图 4-2 各腔室中线虫平均摆动频率随时间的变化

Figure 4-2 Change in contour curvature

## 第五章 总结与展望