

基于微流控芯片和机器视觉的秀丽隐杆线虫表型分析

摘要

目前，环境中化学品的数量日益增多且对人们的身体健康造成严重威胁，如何科学的进行风险评估是公共健康领域的热点与难点。使用细胞或者类器官可进行较高通量的毒性评价与筛选，但研究结果无法代替动物体内实验。传统的动物实验虽然研究结果意义较大，但具有周期长、成本高和涉及动物保护等局限性，难以进行高通量的毒性研究。代表性模式生物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。在线虫的毒理实验中，体长和摆动频率往往作为评价毒性的重要生理指标，通过人工的观察不仅通量低而且还会引入误差。

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，可以实现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，由于线虫通体透明且由 PDMS 制作的微流控芯片也是透明的，线虫的分割是整个系统中的难点，本文首先采用了背景减除的方法进行线虫前景轮廓的提取。实验发现，该方法在较低背景噪声的情况下可以较好的分割线虫。但面对复杂的背景时，该方法的鲁棒性出现显著的下降，导致后续线虫跟踪的丢失率显著增加。针对背景减除方法的不足，本文提出了一种基于深度卷积网络的分割模型。通过实验对比，在我们标注的数据集上分割误差下降了 xx。另外，线虫的跟踪也是一个难点，由于线虫是非刚体其形态变化多种多样。为此，本文提出了一种简单有效的跟踪策略，能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的跟踪。利用跟踪的线虫轮廓，可以计算出线虫的体长、摆动频率和运动速度等生理特征。最后，通过线虫和双氧水的相互作用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；毒性评价；生理监测

PHENOTYPIC ANALYSIS OF C. ELEGANS BASED ON MICROFLUIDIC CHIP AND MACHINE VISION

ABSTRACT

Todo

KEY WORDS: C. elegans, Microfluidics, Toxicity evaluation, Physiological monitoring

目 录

第一章 绪论	1
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	6
1.3.3 机器视觉在线虫研究中的应用	6
1.4 存在的问题	8
1.5 研究内容	8
1.6 论文章节安排	9
第二章 微流控线虫芯片的设计及硬件平台的搭建	10
2.1 引言	10
2.2 材料与方法	10
2.2.1 实验材料与仪器	10
2.2.2 线虫梯度芯片的设计	10
2.2.3 线虫梯度芯片的制作	10
2.3 芯片方案的验证	10
2.4 本章小结	10
第三章 基于深度卷积网络的线虫前景轮廓提取	11
3.1 引言	11
3.2 线虫图像处理总体方案介绍	11
3.3 传统的图像分割方法分类介绍	12
3.3.1 基于像素的图像分割方法	12
3.3.2 基于边缘的图像分割方法	13
3.3.3 基于区域的图像分割方法	13
3.3.4 交互式的图像分割方法	13
3.3.5 基于背景减除的分割方法	14
3.4 线虫轮廓前景分割卷积网络的设计及模型评估	14

3.4.1	数据集的制作	14
3.4.2	条件随机场模型在分割任务中的应用	16
3.4.3	网络结构的设计	16
3.4.4	网络模型的训练	16
3.4.5	实验结果分析	16
3.5	本章小结	16
第四章	基于深度卷积网络的线虫轮廓解析	17
4.1	引言	17
4.2	卷积神经网络介绍	17
4.2.1	卷积层	18
4.2.2	池化层	19
4.2.3	Dropout 层	19
4.2.4	Batch Normalization 层	20
4.3	线虫轮廓解析方法介绍	21
4.3.1	随机点的产生	23
4.3.2	SingleOut-Net 网络输出后处理	23
4.4	SingleOut-net 网络设计及模型评估	24
4.4.1	数据集的制作	24
4.4.2	网络结构的设计	25
4.4.3	损失函数及评价指标	26
4.4.4	网络的训练	27
4.4.5	测试结果与分析	28
4.5	本章小结	29
第五章	线虫的特征提取和急性氧化应激实验	30
5.1	引言	30
5.2	线虫轮廓的跟踪	30
5.3	线虫的特征提取	32
5.3.1	线虫轮廓中间脊线提取	32
5.3.2	身体弯曲角度的计算以及摆动频率的估计	32
5.4	线虫的氧化急性应激实验	33
5.5	本章小结	33
第六章	总结与展望	34

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，对数量巨大且种类复杂的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。此外，人类与环境化合物的接触方式多为低剂量下的复合暴露，如何实现高通量的毒性评价和科学的风险评估成为研究的热点与难点。

自2005年起，美国启动了21世纪毒理学研究计划（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。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

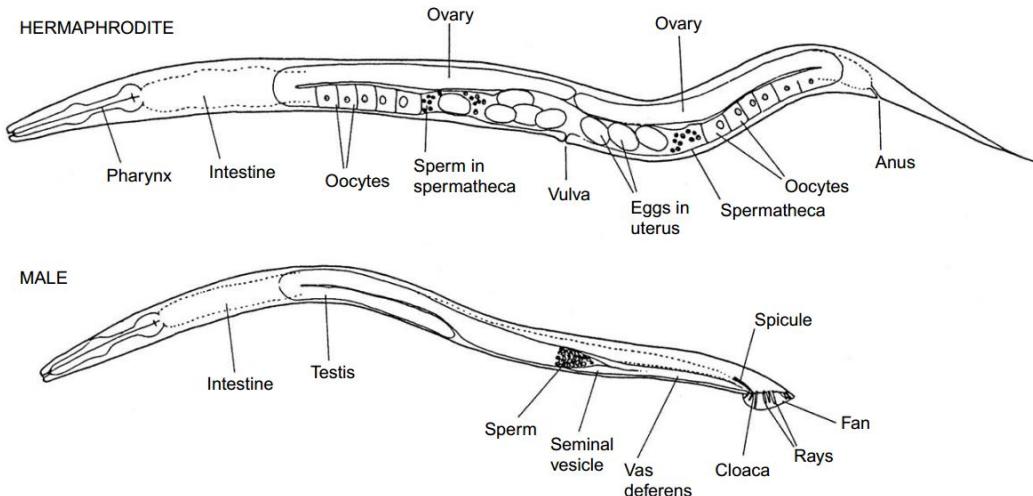


图 1-1 秀丽隐杆线虫解剖图

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

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，但雌雄通体的线虫有两个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。如果雄性体线虫交配产蛋量将会提高，并且有利于基因的重新组合。在25C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线虫具有如下技术优势：

- 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。
- 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。
- 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。

- 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。
- 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。

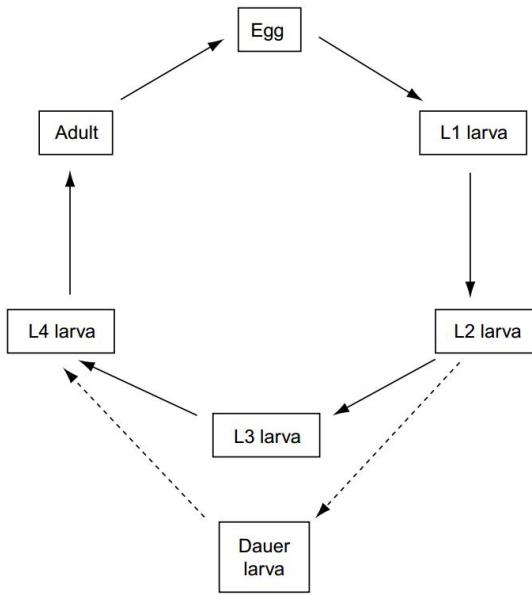


图 1-2 秀丽隐杆线虫生命周期

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。利用微流控技术研究线虫具有的优势有：*a)* 利用微流控芯片可以实现在细胞尺度上对线虫的操纵。*b)* 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。*c)* 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，实现线虫的分选。*d)* 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，包括线虫的分选、线虫固定与成像以及介绍微流控在毒理学实验中的研究进展。最后介绍近年来线虫图像处理的研究进展。

1.3.1 微流控芯片线虫操控方法

在线虫的实验中，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^[1]完成的，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不能实现对单个线虫的跟踪和成像。

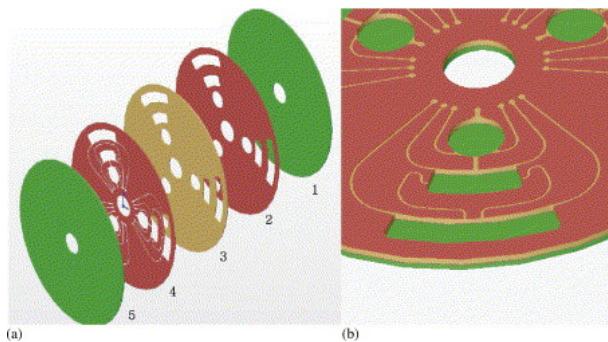


图 1-3 CD 状线虫培养芯片
Figure 1-3 The life cycle of *C.elegans*.

由于以上的局限，Hulme 等^[2]设计了一款新的芯片。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^[3-6]。Krajniak 和 Lu^[4]开发了一款集成微流控芯片，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)下呈现较高地粘性，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。

1.3.1.2 线虫的固定

线虫的固定对于线虫神经元成像与显微手术^[7]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固定线虫的方法。Chokshi 等人^[8]开发了一款简单的微流控芯片可以有效的固定线虫，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，从而使中间的 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。

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^[3]运用 FP127 凝胶液滴将 L1 的幼虫包裹，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^[9]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^[10, 11]，通过对线虫的存活率和子代的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。

1.3.1.3 线虫的分选

目前，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。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、运动能力以及电生理学特性的不同等。Rezai^[12]等人利用线虫的趋电性设计了一种单层微流控芯片，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。Casadevall^[13]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^[14]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^[15, 16]，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^[17]开发了一款包含 48 个平行阵列腔室的微流控芯片，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^[18]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^[19, 20]。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，该芯片由上层液路层、中间气路控制层和底层玻璃组成。液路层有 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^[21]。

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^[22]设计一个可以在较低的倍率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^[23]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^[24]。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^[25] 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^[26] 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个状态线虫轮廓运动的速度，结合当前状态下线虫轮廓的位置可以计算出下个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图片上线虫的轮廓重合。Fontaine 和 Roussel 等人^[27, 28] 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^[29] 提出一种基于统计生成式方法可以识别复杂的姿态。

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^[30] 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的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^[31] 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，并对其进行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^[32] 提出了一种新的计算特征的方法。通过将计算同一个线虫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^[33]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以存储。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，但大规模的片上浓度梯度的形成依然面临问题。目前很多的微流控芯片的片上浓度形成“依赖”圣诞树“结构的浓度生成器”，这种结构存在样品消耗大以及需要精确的流阻设计和流速调节等问题。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。另一方面，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。在国内除本课题外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. 本文首先搭建了一个微流控芯片的硬件平台，可以实现线虫的简单培养并可以快速的形成多个浓度梯度，为线虫在给定化合物下的暴露创造一个精确的微环境。
2. 为了实现对线虫生理指标的实时监测，本文提出一种集线虫轮廓分割、轮廓解析、轮廓跟踪以及特征提取的框架并以 python 作为主要的开发语言加以实现。
3. 针对微流控背景下线虫轮廓分割很容易受到噪声干扰以及鲁棒性等问题，由于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运用在线

虫轮廓分割的任务，结构显示与基于背景减除的轮廓提取算法相比，像素误差下降了 XX。

4. 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和条件生成对抗网络的网络架构。结果显示，这种方法可以有效的解析出线虫相互纠缠时每个线虫的轮廓。

1.6 论文章节安排

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

2.2 材料与方法

2.2.1 实验材料与仪器

2.2.2 线虫梯度芯片的设计

2.2.3 线虫梯度芯片的制作

2.3 芯片方案的验证

2.4 本章小结

第三章 基于深度卷积网络的线虫前景轮廓提取

3.1 引言

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提取具有如下优势：

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，这种方法不依赖超参数的选取，具有很好的鲁棒性。

3.2 线虫图像处理总体方案介绍

线虫图像处理的总体流程如图3-1所示，共分成 4 个阶段。第一个阶段是从采集到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。这一阶段的输出为一幅二值化的图像，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从而跟踪丢失。第二个阶段的任务主要是对多个线虫相交的轮廓进行解析，从而得

到单个线虫的轮廓。第三个阶段的线虫跟踪是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。第四个阶段主要是利用跟踪到的线虫轮廓计算出线虫体长、运动速度等信息。本章主要介绍线虫图像处理流程中第一个阶段的内容。

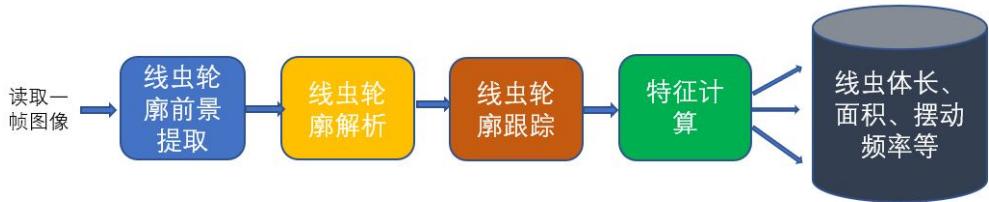


图 3-1 线虫图像处理总体流程图

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

3.3 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。

3.3.1 基于像素的图像分割方法

基于像素的阈值分割方法通过利用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和分类。其中 OSTU 于 1979 年提出了基于最大类间方差的阈值分割方法，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方法外，还有最大熵阈值法、双峰法、P 参数法等。

与通过阈值将像素分为多个类别的阈值分割方法不同，基于像素聚类的分割是通过对像素的灰度值或者特征向量的聚类实现图像分割。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的特征向量里。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算法以消除这些缺陷。常用的后处理算法有区域生长、像素连接和一些基于规则的算法等。常见的聚类算法有 K 均值聚类算法和以模糊集合为基础的 Fuzzy cmeans(FCM) 聚类聚类算法等。

3.3.2 基于边缘的图像分割方法

基于边界的图像分割方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** 算子常被用于计算图像的一阶梯度信息。**Laplacian** 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。

3.3.3 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。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分割成两个或者更多的小区域。另外，区域分裂的分割方法可能使目标轮廓产生孔洞。

3.3.4 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^[34] 和基于曲线演化的活动轮廓分割方法。**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，用户还用画刷将某个区域标记为前景或背景，然后让算法继续迭代以改善分割的性能。另一方面，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的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分割方法。

3.3.5 基于背景减除的分割方法

3.4 线虫轮廓前景分割卷积网络的设计及模型评估

3.4.1 数据集的制作

3.4.1.1 数据集的采集

图3-2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录制。

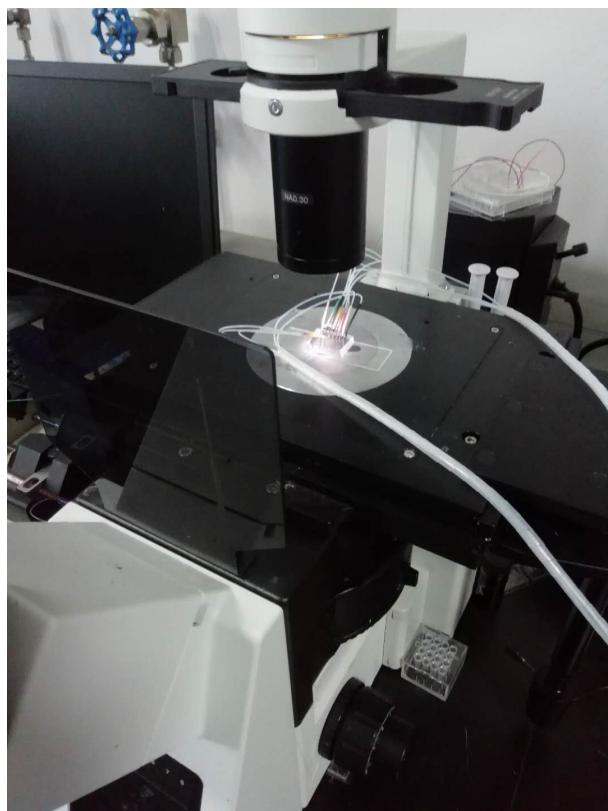


图 3-2 图像采集系统图

Figure 3-2 Change in contour curvature

3.4.1.2 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的标注工具(如：Labelme¹和 Ratesnake²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，因此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。通过这种方法，本文制作了一个包含 211 个样本的数据集，图3-3是数据集部分示例。整个数据集按照 8 : 2 的比例将数据集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。

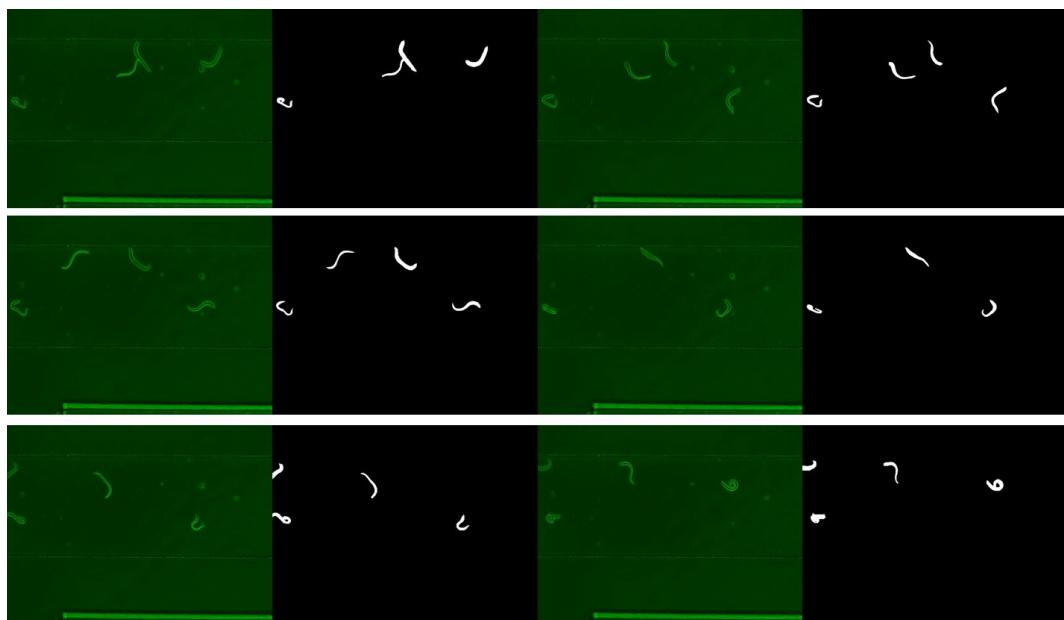


图 3-3 数据集示例

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

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

²<https://is-innovation.eu/ratsnake/>

3.4.2 条件随机场模型在分割任务中的应用

3.4.3 网络结构的设计

3.4.4 网络模型的训练

3.4.5 实验结果分析

3.5 本章小结

第四章 基于深度卷积网络的线虫轮廓解析

4.1 引言

多线虫跟踪问题中，多线虫轮廓之间相互遮挡是造成线虫跟踪丢失的主要原因，是实现线虫长期跟踪的关键。如图4-1所示，是一张经过前景轮廓提取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。虽然人眼可以很轻松的辨别出图中所有单个线虫的轮廓，但自动化地线虫轮廓解析却十分困难，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。本文提出了一种基于深度卷积网络的线虫轮廓解析的方法尝试解决这一问题，并探讨了多种网络结构的设计对识别性能的影响。

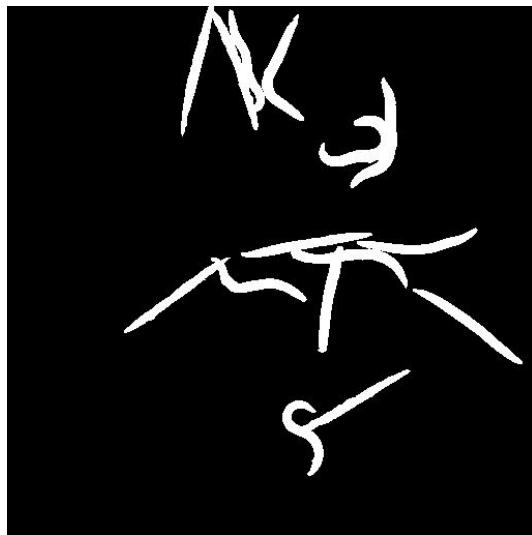


图 4-1 多线虫轮廓相互遮挡图片

Figure 4-1 Change in contour curvature

4.2 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 (Cy-

berneticsim) 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) 利用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，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的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^[35]，卷积操作类似于图像处理中的空间滤波，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4-2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^[36]、VGG 网络^[37]、GoogLeNet 网络^[38] 和 ResNet 网络^[39])。使卷积网络在许多计算机视觉任务中大放异彩。

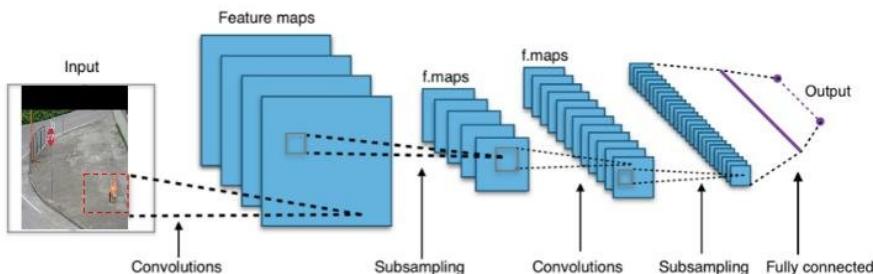


图 4-2 卷积网络的经典结构

Figure 4-2 A typical architecture of CNN

4.2.1 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4-3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4-1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。图4-3中

$i = 6, p = 1, k = 3, s = 2$ 根据公式4-1 可以得到 $o = 3$ 。

$$o = \lfloor \frac{i + 2p - k}{s} \rfloor + 1 \quad (4-1)$$

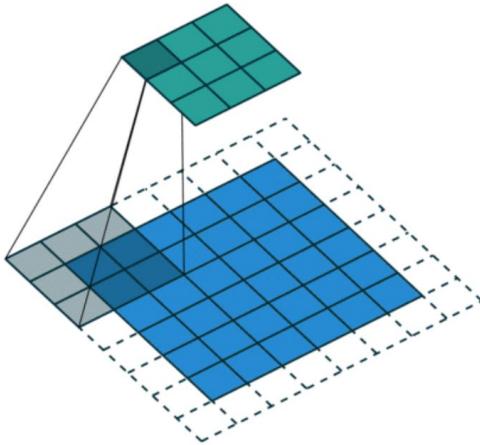


图 4-3 卷积操作示意图

Figure 4-3 A typical architecture of CNN

4.2.2 池化层

池化层是一种对输入张量进行降采样的操作，由于其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，其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4-4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。

4.2.3 Dropout 层

为了解决神经网络过拟合的问题，Srivastava 和 Hinton 等人^[40]于 2014 年提出 dropout 方法，有效的提高了网络的泛化能力。如图4-5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网络的权值以一定的比例缩放。

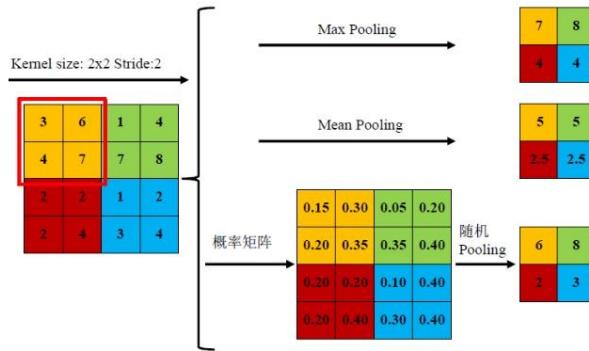


图 4-4 卷积操作示意图

Figure 4-4 A typical architecture of CNN

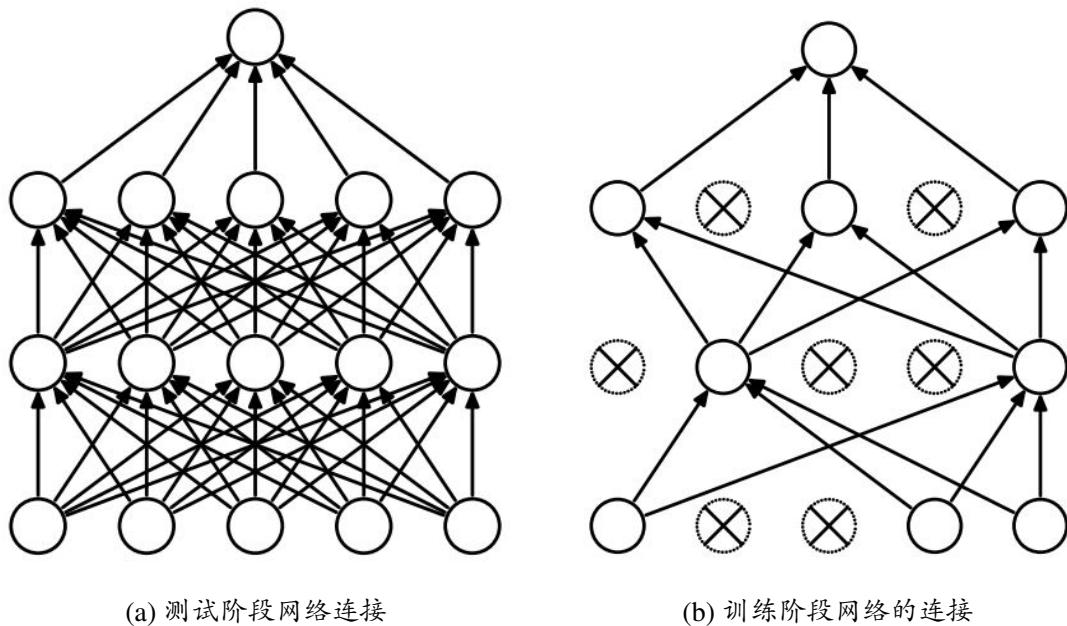


图 4-5 线虫弯曲角度和摆动频率的变换

Figure 4-5 An EPS and PDF demo with subcaptionbox

4.2.4 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^[41]，有效的解决了内部方差偏移 (Internal Covariate Shift) 的问题。内部方差偏移指的是在网络训练的过程中，隐藏层的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 sigmoid 函数等) 在 0 值附近的梯度是最大的，因此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，还降低了对初

始学习率选取的敏感度。

4.3 线虫轮廓解析方法介绍

本文提出的线虫轮廓解析方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 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 个线虫的图片， R 表示渲染函数， ξ 表示随机噪声。 $I(E_0) = R(\{e_0\}; \xi)$ 表示由中间线虫渲染得到的图片。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。

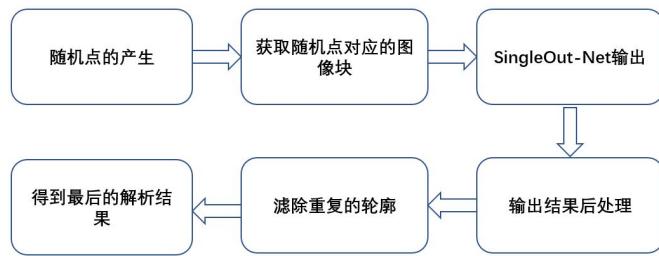


图 4-6 多线虫轮廓相互遮挡图片

Figure 4-6 Change in contour curvature

如图4-6是本文提出的线虫轮廓解析方法的主要流程。首先假定从前景提取的步骤中已经得到包含多线虫轮廓的二值化图像如图4-1所示，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抠出固定尺寸的图像块，将这些图像块输入到训练好的 SingleOut-Net 网络从而得到中央线虫的轮廓。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包含同一个线虫的多个轮廓，在原图像坐标中，同一个线虫的多个轮廓是重合的。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4-1描述线虫轮廓解析的整个算法实现。

算法 4-1 线虫轮廓解析算法

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 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
```



图 4-7 多线虫轮廓相互遮挡图片

Figure 4-7 Change in contour curvature

4.3.1 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，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，找出原图中所有的轮廓，然后在每个轮廓的边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4-7所示。

4.3.2 SingleOut-Net 网络输出后处理

图4-8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 网络成功的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，当两个轮廓的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4-9 所示。

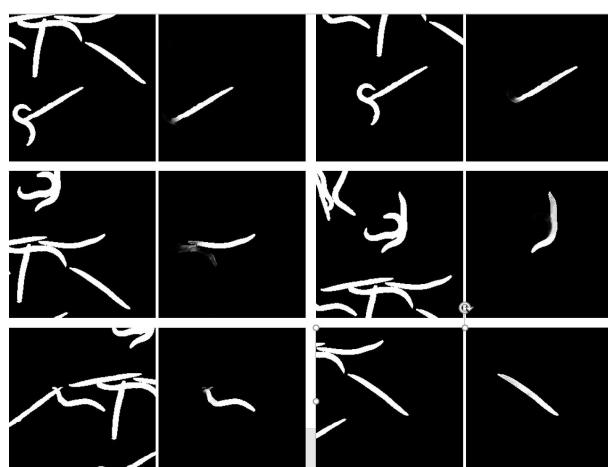


图 4-8 多线虫轮廓相互遮挡图片

Figure 4-8 Change in contour curvature



图 4-9 多线虫轮廓相互遮挡图片

Figure 4-9 Change in contour curvature

4.4 SingleOut-net 网络设计及模型评估

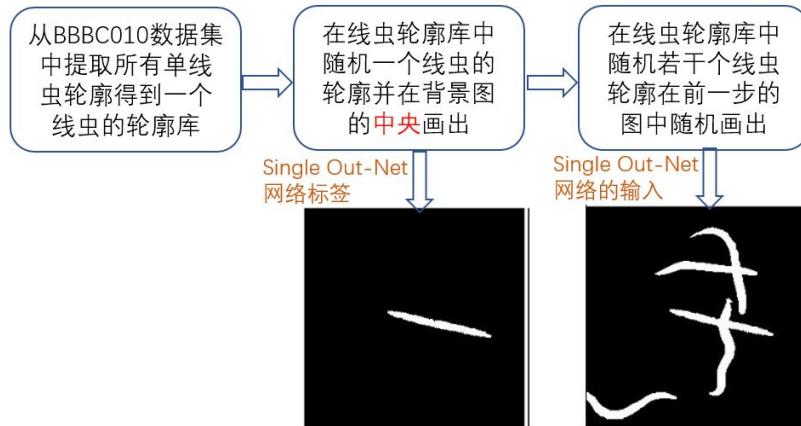


图 4-10 多线虫轮廓相互遮挡图片

Figure 4-10 Change in contour curvature

4.4.1 数据集的制作

SingleOut-Net 网络的训练需要大量的标定的数据集，本文采用了人工生成的数据集来训练网络的参数。数据集的制作流程如图4-10所示。BBBC010 数据集^[42]中包含 1407 张单线虫轮廓的图像。首先，我们从 BBBC010 数据集中提取所有单线虫的轮廓得到一个单线虫的轮廓库。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 256×256 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 网络的输入以及对应的标签。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输出归一化到 $0 \sim 1$ 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。

4.4.2 网络结构的设计

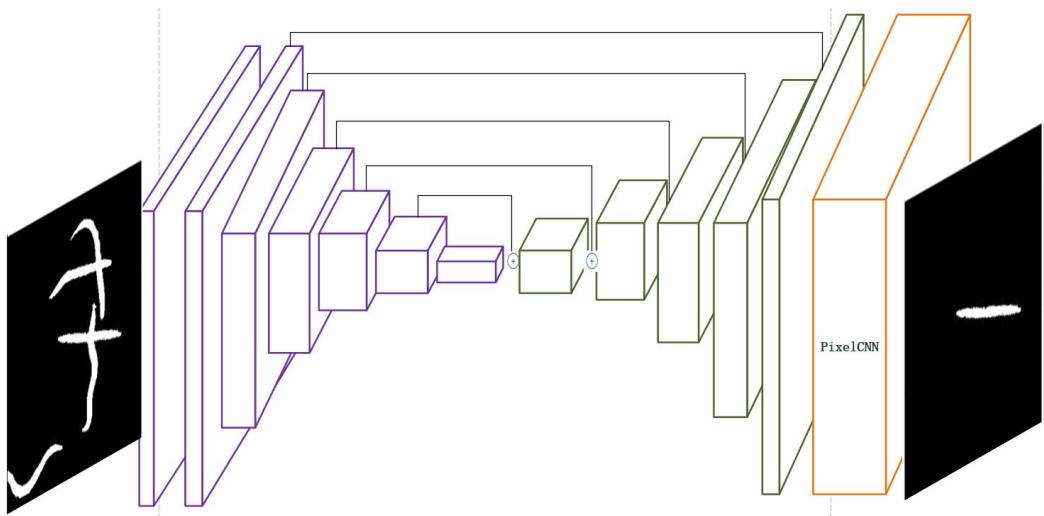


图 4-11 多线虫轮廓相互遮挡图片

Figure 4-11 Change in contour curvature

如图4-11所示是 SingleOut-Net 的网络结构。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^[43] 的卷积模块在整个网络架构中相当于一个编码器，但与 U-net 网络使用的卷积模块不同，本文使用了如图4-12所示的残差连接模块^[44]，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。第一个模块包含两条路径，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，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 2×2 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^[45] 具有克服梯度消失和加快网络的收敛等优势，这里使用了 Relu 激活函数。在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 2×2 步长为 2。如图4-11所示，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。第二个模块为一个 PixelCNN 网络^[46] 模块作为解码器。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 网络中作为解码器来生成线虫图像。最后通过一个 1×1 的卷积将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。

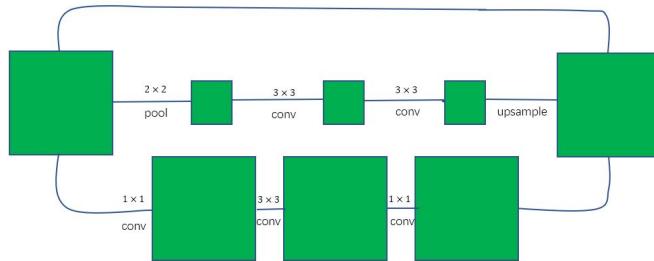


图 4-12 多线虫轮廓相互遮挡图片

Figure 4-12 Change in contour curvature

4.4.3 损失函数及评价指标

4.4.3.1 损失函数

在对神经网络进行训练之前，首先需要确定网络训练的损失函数(也即目标函数)。神经网络通过不断地调整网络的模型参数使得训练损失不断减小。由上一小节可知，网络的最后一层输出是一张概率图设为 p ， y 为标签图(只包含 0 和 1)。公式4-2表示网络训练的交叉熵损失。

$$\text{entropy_loss} = \sum_{i,j} y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij}) \quad (4-2)$$

4.4.3.2 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4-3表示。

$$pixel_cls_error = 1 - \frac{number_of_corrected_classified_pixels}{total_number_of_pixels} \quad (4-3)$$

4.4.4 网络的训练

根据4.4.1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。另外为了比较不同的网络架构的性能，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。

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选取时加以考虑。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杂度和实时性，表4-1列出了本文中所有网络模型运行的硬件平台。

表 4-1 算法运行的实验平台

Table 4-1 A Table

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

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。网络权值的更新往往是基于小批量样本而不是单个样本，其中 batchsize 是一个重要参数，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，会导致网络模型陷入局部最小值点。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^[47] 和 Adam 算法^[48] 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。训练 40 个 epoch（每个 epoch 包含 1000 个 batch）后网络已经完全收敛，图 4-13 表示损失函数随 epoch 数的增加而下降。

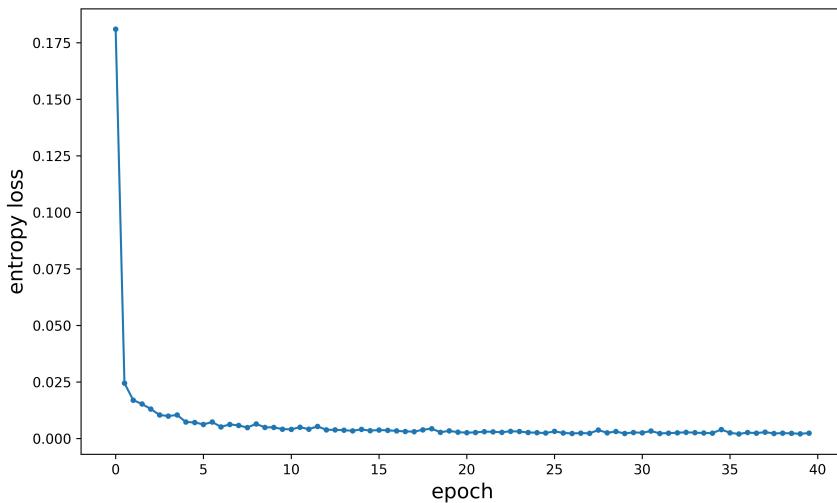


图 4-13 网络训练过程中损失函数的变化

Figure 4-13 Change in contour curvature

为了观察网络训练过程中 SingleOut-Net 网络性能的变化，本文对网络每隔 100 次迭代进行一次测试，图 4-14 显示了网络训练过程中对模型测试的结果。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。

4.4.5 测试结果与分析

在 4.4.2 节，本文介绍了 SingleOut-Net 的网络架构，由两个模块构成编解码器结构。后端模块是一个 pixelCNN，相当于一个解码器的作用。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

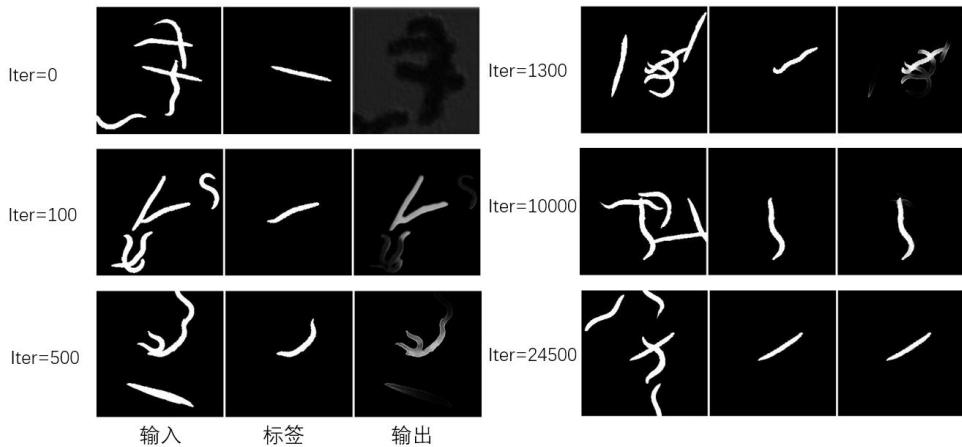


图 4-14 网络训练过程中模型测试结果

Figure 4-14 Change in contour curvature

了 pixelCNN 网络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，所以可以将 pixelCNN 模块移去，然后用一个 1×1 的卷积将编码器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一，4.4.2节介绍的网络架构命名为模型二，两个模型的性能比较如表4-2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，模型大小只有少量增加。

表 4-2 网络模型性能的比较

Table 4-2 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

4.5 本章小结

第五章 线虫的特征提取和急性氧化应激实验

5.1 引言

5.2 线虫轮廓的跟踪

由于线虫通体透明，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提取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5-1和公式5-2可以计算出轮廓的重心坐标。

$$m_{ji} = \sum_{x,y} I_{x,y} x^i y^j \quad (5-1)$$

$$\vec{x} = \frac{m_{10}}{m_{00}}, \quad \vec{y} = \frac{m_{01}}{m_{00}} \quad (5-2)$$

假设当前帧有 n 个轮廓，上一帧图像有 m 个轮廓，由每个轮廓的重心坐标可以得到一个 $n \times m$ 的距离矩阵用公式5-3表示。

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{bmatrix} \quad (5-3)$$

矩阵中 d_{ij} 表示当前帧中的第 i 个轮廓的重心到上一帧中第 j 个轮廓的重心之间的距离。通过公式5-4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。

$$index(i) = \arg \min_j d_{ij} \quad (5-4)$$

但事实上由于轮廓分割的不完美以及图像噪声的影响，这一策略往往失效应。因此，通过最近邻搜索的方式来实现线虫的跟踪要满足以下的约束条件，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5-2 是描述了线虫跟踪算法的实现思路。图5-1表示跟踪的结果。

- 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的重心之间的距离应该小于一个阈值。

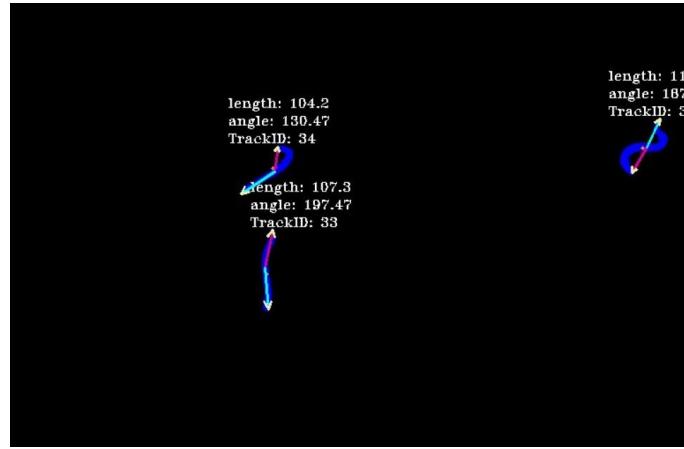


图 5-1 跟踪的结果

Figure 5-1 The tracking result

算法 5-1 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 5-2 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  $frame_i index = 1 \rightarrow Worm\_Data.length - 1$  do
4:      $PreFrame\_WormData \leftarrow Worm\_Data[frame_i index - 1]$ 
5:     for  $worm\_index = 0 \rightarrow Worm\_Data[frame_i index].length - 1$  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  $dist\_array[min\_index] < \sigma$ 
11:        then
12:           $cur\_worm.trackID \leftarrow Nearest\_worm.trackID$ 
13:        else
14:           $cur\_worm.trackID \leftarrow GetNewTrackID()$ 

```

5.3 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，在其轮廓上采样 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估计以及运动速度的计算。

5.3.1 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，在轮廓上的任意一点其两边都可以找到一个单位边长度的相邻点，这三点所成角的补角的倒数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变换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，则这条曲线上将会有两个波峰如图5-2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。

5.3.2 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^[49]。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°C 左右振荡如图??，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5-5得出，图??表示线虫摆动频率随时间的变换。

$$f = \frac{\text{frame_rate} * n}{2 * \Delta t} \quad (5-5)$$

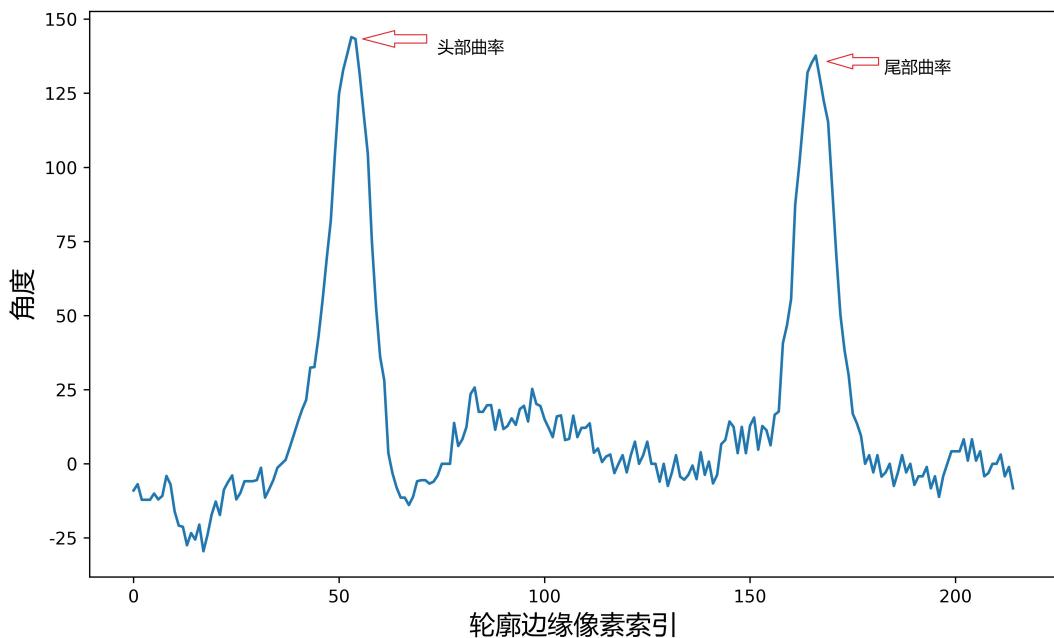


图 5-2 轮廓曲率的变化

Figure 5-2 Change in contour curvature

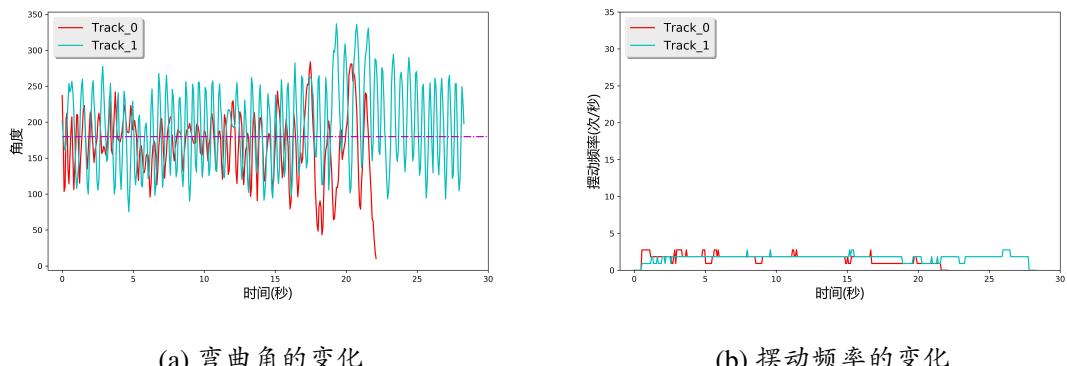


图 5-3 线虫弯曲角度和摆动频率的变换

Figure 5-3 An EPS and PDF demo with subcaptionbox

5.4 线虫的氧化急性应激实验

5.5 本章小结

第六章 总结与展望

参考文献

- [1] KIM N, DEMPSEY C M, ZOVAL J V, et al. Automated microfluidic compact disc (CD) cultivation system of *Caenorhabditis elegans*[J]. Sensors & Actuators B Chemical, 2007, 122(2): 511–518.
- [2] HULME S E, SHEVKOPLYAS S S, MCGUIGAN A P, et al. Lifespan-on-a-chip: microfluidic chambers for performing lifelong observation of *C. elegans*[J]. Lab on A Chip, 2010, 10(5): 589–597.
- [3] AUBRY G, MEI Z, LU H. Hydrogel-droplet microfluidic platform for high-resolution imaging and sorting of early larval *Caenorhabditis elegans*[J]. Lab on A Chip, 2015, 15(6): 1424–1431.
- [4] KRAJNIAK J, LU H. Long-term high-resolution imaging and culture of *C. elegans* in chip-gel hybrid microfluidic device for developmental studies[J]. Lab on A Chip, 2010, 10(14): 1862–1868.
- [5] WEN H, YU Y, ZHU G, et al. A droplet microchip with substance exchange capability for the developmental study of *C. elegans*[J]. Lab on A Chip, 2015, 15(8): 1905.
- [6] CORNAGLIA M, KRISHNAMANI G, MOUCHIROUD L, et al. Automated longitudinal monitoring of *in vivo* protein aggregation in neurodegenerative disease *C. elegans* models.[J]. Molecular Neurodegeneration, 2016, 11(1): 1–13.
- [7] GOKCE S K, GUO S X, GHORASHIAN N, et al. A fully automated microfluidic femtosecond laser axotomy platform for nerve regeneration studies in *C. elegans*.[J]. Plos One, 2014, 9(12): e113917.
- [8] CHOKSHI T V, BENYAKAR A, CHRONIS N. CO₂ and compressive immobilization of *C. elegans* on-chip.[J]. Lab on A Chip, 2009, 9(1): 151–157.
- [9] CORNAGLIA M, MOUCHIROUD L, MARETTE A, et al. An automated microfluidic platform for *C. elegans* embryo arraying, phenotyping, and long-term live imaging[J]. Sci Rep, 2015, 5: 10192.

- [10] LEE H, KIM S A, COAKLEY S, et al. A multi-channel device for high-density target-selective stimulation and long-term monitoring of cells and subcellular features in *C. elegans*[J]. *Lab on A Chip*, 2014, 14(23): 4513–4522.
- [11] HULME S E, SHEVKOPLYAS S S, APFELD J, et al. A microfabricated array of clamps for immobilizing and imaging *C. elegans*[J]. *Lab on A Chip*, 2007, 7(11): 1515–1523.
- [12] REZAI P, SALAM S, SELVAGANAPATHY P R, et al. Electrical sorting of *Caenorhabditis elegans*[J]. *Lab on A Chip*, 2012, 12(10): 1831.
- [13] CASADEVALL I S X, GEIER F M, LEROI A M, et al. High-throughput age synchronisation of *Caenorhabditis elegans*[J]. *Chemical Communications*, 2011, 47(35): 9801.
- [14] DONG L, CORNAGLIA M, LEHNERT T, et al. Versatile size-dependent sorting of *C. elegans* nematodes and embryos using a tunable microfluidic filter structure[J]. *Lab on a Chip*, 2016, 16(3): 574–585.
- [15] BAUMEISTER R, GE L. The worm in us - *Caenorhabditis elegans* as a model of human disease.[J]. *Trends in Biotechnology*, 2002, 20(4): 147–148.
- [16] SONNHAMMER E L, DURBIN R. Analysis of protein domain families in *Caenorhabditis elegans*.[J]. *Genomics*, 1997, 46(2): 200–216.
- [17] CHUNG K, ZHAN M, SRINIVASAN J, et al. Microfluidic chamber arrays for whole-organism behavior-based chemical screening[J]. *Lab on A Chip*, 2011, 11(21): 3689.
- [18] YANG J, CHEN Z, CHING P, et al. An integrated microfluidic platform for evaluating *in vivo* antimicrobial activity of natural compounds using a whole-animal infection model[J]. *Lab on A Chip*, 2013, 13(17): 3373.
- [19] DERTINGER S K W, CHIU D T, JEON N L, et al. Generation of Gradients Having Complex Shapes Using Microfluidic Networks[J]. *Analytical Chemistry*, 2001, 73(6): 1240–1246.
- [20] JEON N L, DERTINGER S K W, CHIU D T, et al. Generation of Solution and Surface Gradients Using Microfluidic Systems[J]. *Langmuir*, 2000, 16(22): 8311–8316.
- [21] 朱国丽, 尹方超, 王丽, 等. 基于微流控芯片的高糖对模式生物线虫寿命影响及白藜芦醇昔保护性作用考察[J]. 色谱, 2016, 34(2): 140–145.

- [22] DE B M, BARGMANN C I. Natural variation in a neuropeptide Y receptor homolog modifies social behavior and food response in *C. elegans*.[J]. *Cell*, 1998, 94(5): 679–689.
- [23] BAEK J H, COSMAN P, FENG Z, et al. Using machine vision to analyze and classify *Caenorhabditis elegans* behavioral phenotypes quantitatively[J]. *Journal of Neuroscience Methods*, 2002, 118(1): 9–21.
- [24] YEMINI E. High-throughput, single-worm tracking and analysis in *Caenorhabditis elegans*[D]. University of Cambridge, 2013.
- [25] SWIERCZEK N A, GILES A C, RANKIN C H, et al. High-throughput behavioral analysis in *C. elegans*[J]. *Nature methods*, 2011, 8(7): 592.
- [26] RESTIF C, METAXAS D. Tracking the swimming motions of *C. elegans* worms with applications in aging studies[C] // International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer. [S.l.]: [s.n.], 2008: 35–42.
- [27] FONTAINE E, BARR A H, BURDICK J W. Model-based tracking of multiple worms and fish[C] // ICCV Workshop on Dynamical Vision. [S.l.]: [s.n.], 2007.
- [28] ROUSSEL N, SPRENGER J, TAPPAN S J, et al. Robust tracking and quantification of *C. elegans* body shape and locomotion through coiling, entanglement, and omega bends[C] // Worm. Vol. 3. 4. Taylor & Francis. [S.l.]: [s.n.], 2014: e982437.
- [29] NAGY S, GOESSLING M, AMIT Y, et al. A generative statistical algorithm for automatic detection of complex postures[J]. *PLoS computational biology*, 2015, 11(10): e1004517.
- [30] YEMINI E, JUCIKAS T, GRUNDY L J, et al. A database of *Caenorhabditis elegans* behavioral phenotypes[J]. *Nature methods*, 2013, 10(9): 877.
- [31] STEPHENS G J, JOHNSON-KERNER B, BIALEK W, et al. Dimensionality and dynamics in the behavior of *C. elegans*[J]. *Plos Computational Biology*, 2008, 4(4): e1000028.
- [32] RESTIF C, IBÁÑEZVENTOSO C, VORA M M, et al. CeleST: computer vision software for quantitative analysis of *C. elegans* swim behavior reveals novel features of locomotion.[J]. *PLoS Computational Biology*, 10,7(2014-7-17), 2014, 10(7): e1003702.

- [33] JAVER A, CURRIE M, LEE C W, et al. An open-source platform for analyzing and sharing worm-behavior data[J]. *Nature methods*, 2018, 15(9): 645.
- [34] ROTHER C, KOLMOGOROV V, BLAKE A. Grabcut: Interactive foreground extraction using iterated graph cuts[C]// ACM transactions on graphics (TOG). Vol. 23. 3. ACM. [S.l.]: [s.n.], 2004: 309–314.
- [35] LE CUN Y, JACKEL L, BOSER B, et al. Handwritten digit recognition: Applications of neural network chips and automatic learning[J]. *IEEE Communications Magazine*, 1989, 27(11): 41–46.
- [36] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]// Advances in neural information processing systems. [S.l.]: [s.n.], 2012: 1097–1105.
- [37] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. ArXiv preprint arXiv:1409.1556, 2014.
- [38] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.]: [s.n.], 2015: 1–9.
- [39] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition[J]. 2015: 770–778.
- [40] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *The Journal of Machine Learning Research*, 2014, 15(1): 1929–1958.
- [41] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. ArXiv preprint arXiv:1502.03167, 2015.
- [42] V L, KL S, AE C. Annotated high-throughput microscopy image sets for validation.[J]. *Nature Methods*, 2012, 9(7): 637.
- [43] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]// International Conference on Medical image computing and computer-assisted intervention. Springer. [S.l.]: [s.n.], 2015: 234–241.
- [44] CHU X, YANG W, OUYANG W, et al. Multi-context attention for human pose estimation[J]. ArXiv preprint arXiv:1702.07432, 2017, 1(2).

- [45] XU B, WANG N, CHEN T, et al. Empirical evaluation of rectified activations in convolutional network[J]. ArXiv preprint arXiv:1505.00853, 2015.
- [46] Van den OORD A, KALCHBRENNER N, ESPEHOLT L, et al. Conditional image generation with pixelcnn decoders[C]// Advances in Neural Information Processing Systems. [S.l.]: [s.n.], 2016: 4790–4798.
- [47] ZEILER M D. ADADELTA: an adaptive learning rate method[J]. ArXiv preprint arXiv:1212.5701, 2012.
- [48] KINGA D, ADAM J B. A method for stochastic optimization[C]// International Conference on Learning Representations (ICLR). Vol. 5. [S.l.]: [s.n.], 2015.
- [49] WANG D, XING X. Assessment of locomotion behavioral defects induced by acute toxicity from heavy metal exposure in nematode *Caenorhabditis elegans*[J]. 环境科学学报(英文版), 2008, 20(9): 1132–1137.