UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Desarrollo de técnicas avanzadas de seguimiento de posturas para reconocimiento de comportamientos de *C. elegans*

Development of advanced posture-tracking techniques for behaviour recognition in *C. elegans*

TESIS DOCTORAL
Pablo Emmanuel Layana Castro

Director:
Antonio José Sánchez Salmerón

Julio 2023

Fundings and grants

This research was supported by Ministerio de Ciencia, Innovación y Universidades [RTI2018-094312-B-I00 (European FEDER funds); FPI PRE2019-088214], and also was supported by Universitat Politècnica de València ["Funding for open access charge: Universitat Politècnica de València"].

The author received a scholarship from the grant: Ayudas para contratos predoctorales para la formación de doctores 2019.

Acknowledgments

Durante todo el recorrido de esta tesis han habido muchos momentos de alegría, así como también momentos difíciles, sin embargo, llegar hasta el final no hubiese sido posibles sin la ayuda de muchas personas a las cuales quiero mostrar mi gratitud.

En primer lugar, quiero agradecer a mi tutor de tesis, Antonio José Sánchez Salmerón, por todo su conocimiento y enseñanzas durante todo el transcurso de la tesis. Gracias por depositar su confianza en mi para el desarrollo de esta tesis, sin su apoyo no hubiese podido conseguir la ayuda del contrato predoctoral que ha financiado esta investigación. También quiero agradecer a la agencia estatal de investigación, ministerios de ciencia e innovación del gobierno de españa por la ayudas para contratos predoctorales para la formación de doctores 2019, beca FPI PRE2019-088214 y al proyecto «RETOS INVESTIGACIÓN» CONOCE (Ref. RTI2018-094312-B-I00, 2019-202) por permitirme ser parte del proyecto.

Un profundo agradecimiento a mi hermano Pedro, sin su ayuda, consejos y respaldo no hubiese sido posible empezar y terminar esta gran hazaña. También quiero agradecer a mi madre María, a mi hermana Marcela, y a toda mi familia incluyendo a la pequeña Carlie y a mi amiga Ariana por su apoyo incondicional, aunque la distancias nos separó en muchas ocaciones, siempre estuvieron presente en cada paso de esta tesis. Muchas gracias por sus consejos y palabras de motivación fueron de mucha ayuda, los quiero mucho.

Gracias a mis amigos David, Isabel, Laura, Cris, Alberto, MariLoli, Vicente, Alicia, Guille, Clara, Isa, César y Alex, su apoyo emocional y amistad me han dado fuerza de seguir motivado durante todo el doctorado. Gracias por preocuparse y cuidar de mi en todo momento. También quiero agradecer a mis amigos Eli, José Luis, Christian y en especial a mi amiga Miry, sus consejos y logros durante su camino de doctorado me han inspirado, y han sido un ejemplo a seguir, estoy muy orgulloso de ellos. Gracias a todos por los momentos que compartimos y las experiencias vividas, es algo que atesoro con mucho cariño. Sus palabras de motivación y consejos me han ayudado en los momentos más difíciles.

Gracias a mis compañeros de laboratorio del grupo de investigación de visión artificial y robótica, Antonio, Joan, Francisco y Ernesto por toda la ayuda en el desarrollo de esta tesis. Sin su ayuda y colaboración no hubiese sido posible las revisiones y finalización de los papers. Gracias por la experiencia compartida en cada artículo y proyecto del laboratorio.

I would like to thank everyone at TAVERNARAKIS LAB for accepting me into their research group and sharing their knowledge during my research stay on Crete Island in Greece. Their knowledge in biology helped me to understand more about the world of *C. elegans*. My experience in Greece wouldn't have been as much fun without their support and friendship.

También quiero agradecer a la subvención: EU-FEDER Comunitat Valenciana 2014-2020, IDIFEDER/2018/025, por el cluster de ordenadores donde se han entrenado y se continúa entrenando modelos de aprendizaje profundo de trabajos propuestos y trabajos relacionados. También quiero agradecer a Ana Maria Alcalá Dávalos, por todo el trabajo de configuración y puesta en marcha de este cluster, así como también todo su trabajo de TI y de gestión de archivos compartidos/datasets en los servidores de la UPV.

Abstract

The main objective of this thesis is the development of advanced posture-tracking techniques for behavioural recognition of *Caenorhabditis elegans* or *C. elegans*. *C. elegans* is a kind of nematode used as a model organism for the study and treatment of different pathological and neurodegenerative diseases. Their behaviour provides valuable information for the research of new drugs (or healthy food and cosmetic products) in the study of lifespan and healthspan. Today, many of the tests on *C. elegans* are performed manually, i.e. using microscopes to track them and observe their behaviour, or in more modern laboratories using specific software. These programmes are not fully automatic, requiring parameter adjustment. And in other cases, they are programmes for image visualisation where the operator must label the behaviour of each *C. elegans* manually. All this translates into many hours of work, which can be automated using computer vision techniques. In addition to being able to estimate mobility indicators more accurately than a human operator.

The main problem in tracking *C. elegans* postures in Petri dishes is aggregations between nematodes or with noise from the environment. Loss or changes of identity are very common either manually or using automatic/semi-automatic programs. And this problem becomes even more complicated in low-resolution images. Programs that automate these pose-tracking tasks work with computer vision techniques using either traditional image processing techniques or deep learning techniques. Both techniques have shown excellent results in the detection and tracking of *C. elegans* postures. On the one hand, traditional techniques use algorithms/optimizers to obtain the best solution, while deep learning techniques automatically learn features from the training dataset. The problem with deep learning techniques is that they need a dedicated and large dataset to train the models.

The methodology used for the development of this thesis (advanced posture-tracking techniques) falls within the research area of computer vision. It has been approached by exploring both branches of computer vision to solve the posture-tracking problems of *C. elegans* in low-resolution images. The first part, i.e. sections 1 and 2, chapter 2, used traditional image processing techniques to perform posture detection and tracking of *C. elegans*. For this purpose, a new skeletonization technique and two new evaluation criteria were proposed to obtain better posture-tracking, detection, and segmentation results. The next sections of chapter 2 use deep learning techniques, and synthetic image simulation to train models and improve posture detection and prediction results. The results proved to be faster and more accurate compared to traditional techniques. Deep learning methods were also shown to be more robust in the presence of plate noise.

Keywords: *C. elegans*; Lifespan; Healthspan; image processing; traditional techniques; deep learning; detection; skeletonization; multi-tracker; low-resolution images; behavioural simulator; synthetic images.

Resumen

El objetivo principal de esta tesis es el desarrollo de técnicas avanzadas de seguimiento de posturas para reconocimiento de comportamientos del *Caenorhabditis elegans* o *C. elegans*. El *C. elegans* es una clase de nematodo utilizado como organismo modelo para el estudio y tratamientos de diferentes enfermedades patológicas así como neurodegenerativas. Su comportamiento ofrece información valiosa para la investigación de nuevos fármacos (o productos alimenticios y cosméticos saludables) en el estudio de lifespan y healthspan. Al día de hoy, muchos de los ensayos con *C. elegans* se realizan de forma manual, es decir, usando microscopios para seguirlos y observar sus comportamientos o en laboratorios más modernos utilizando programas específicos. Estos programas no son totalmente automáticos, requieren ajuste de parámetros. Y en otros casos, son programas para visualización de imágenes donde el operador debe etiquetar maualmente el comportamiento de cada *C. elegans*. Todo esto se traduce a muchas horas de trabajo, lo cual se puede automatizar utilizando técnicas de visión por computador. Además de poder estimar indicadores de movilidad con mayor precisión que un operador humano.

El problema principal en el seguimiento de posturas de *C. elegans* en placas de Petri son las agregaciones entre nematodos o con ruido del entorno. La pérdida o cambios de identidad son muy comunes ya sea de forma manual o usando programas automáticos/semi-automáticos. Y este problema se vuelve más complicado aún en imágenes de baja resolución. Los programas que automatizan estas tareas de seguimiento de posturas trabajan con técnicas de visión por computador usando técnicas tradicionales de procesamiento de imágenes o técnicas de aprendizaje profundo. Ambas técnicas han demostrado excelentes resultados en la detección y seguimiento de posturas de *C. elegans*. Por un lado, técnicas tradicionales utilizan algoritmos/optimizadores para obtener la mejor solución, mientras que las técnicas de aprendizaje profundo aprenden de forma automática características del conjunto de datos de entrenamiento. El problema con las técnicas de aprendizaje profundo es que necesitan un conjunto de datos dedicado y grande para entrenar los modelos.

La metodología utilizada para el desarrollo de esta tesis (técnicas avanzadas de seguimiento de posturas) se encuadran dentro del área de investigación de la visión artificial. Y ha sido abordada explorando ambas ramas de visión por computador para resolver los problemas de seguimiento de posturas de *C. elegans* en imágenes de baja resolución. La primera parte, es decir, secciones 1 y 2, capítulo 2, utilizó técnicas tradicionales de procesamiento de imágenes para realizar la detección y seguimiento de posturas de los *C. elegans*. Para ello se propuso una nueva técnica de esqueletización y dos nuevos criterios de evaluación para obtener mejores resultados de seguimiento, detección, y segmentación de posturas. Las siguientes secciones del capítulo 2 utilizan técnicas de aprendizaje profundo, y simulación de imágenes sintéticas para entrenar modelos y mejorar los resultados de detección y predicción de posturas. Los resultados demostraron ser más rápidos y más precisos en comparación con técnicas tradicionales. También se demostró que los métodos de aprendizaje profundo son más robustos ante la presencia de ruido en la placa.

Palabras claves: *C. elegans*; Lifespan; Healthspan; procesaiento de imágenes; técnicas tradicionales; aprendizaje profundo; detección; esqueletización, seguidor múltiple; imágenes de baja resolución; simulador de comportamientos; imágenes sintéticas.

Resum

L'objectiu principal d'aquesta tesi és el desenvolupament de tècniques avançades de seguiment de postures per a reconeixement de comportaments del *Caenorhabditis elegans* o *C. elegans*. El *C. elegans* és una classe de nematodo utilitzat com a organisme model per a l'estudi i tractaments de diferents malalties patològiques així com neurodegeneratives. El seu comportament ofereix informació valuosa per a la investigació de nous fàrmacs (o productes alimentosos i cosmètics saludables) en l'estudi de lifespan i healthspan. Al dia de hui, molts dels assajos amb *C. elegans* es realitzen de manera manual, és a dir, usant microscopis per a seguir-los i observar els seus comportaments o en laboratoris més moderns utilitzant programes específics. Aquests programes no són totalment automàtics, requereixen ajust de paràmetres. I en altres casos, són programes per a visualització d'imatges on l'operador ha d'etiquetar maualment el comportament de cada *C. elegans*. Tot això es tradueix a moltes hores de treball, la qual cosa es pot automatitzar utilitzant tècniques de visió per computador. A més de poder estimar indicadors de mobilitat amb major precisió que un operador humà.

El problema principal en el seguiment de postures de *C. elegans* en plaques de Petri són les agregacions entre nematodes o amb soroll de l'entorn. La pèrdua o canvis d'identitat són molt comuns ja siga de manera manual o usant programes automàtics/semi-automàtics. I aquest problema es torna més complicat encara en imatges de baixa resolució. Els programes que automatitzen aquestes tasques de seguiment de postures treballen amb tècniques de visió per computador usant tècniques tradicionals de processament d'imatges o tècniques d'aprenentatge profund. Totes dues tècniques han demostrat excel·lents resultats en la detecció i seguiment de postures de *C. elegans*. D'una banda, tècniques tradicionals utilitzen algorismes/optimizadors per a obtindre la millor solució, mentre que les tècniques d'aprenentatge profund aprenen de manera automàtica característiques del conjunt de dades d'entrenament. El problema amb les tècniques d'aprenentatge profund és que necessiten un conjunt de dades dedicat i gran per a entrenar els models.

La metodologia utilitzada per al desenvolupament d'aquesta tesi (tècniques avançades de seguiment de postures) s'enquadren dins de l'àrea d'investigació de la visió artificial. I ha sigut abordada explorant totes dues branques de visió per computador per a resoldre els problemes de seguiment de postures de *C. elegans* en imatges de baixa resolució. La primera part, és a dir, secció 1 i 2, capítol 2, va utilitzar tècniques tradicionals de processament d'imatges per a realitzar la detecció i seguiment de postures dels *C. elegans*. Per a això es va proposar una nova tècnica de esqueletizació i dos nous criteris d'avaluació per a obtindre millors resultats de seguiment, detecció i segmentació de postures. Les següents seccions del capítol 2 utilitzen tècniques d'aprenentatge profund i simulació d'imatges sintètiques per a entrenar models i millorar els resultats de detecció i predicció de postures. Els resultats van demostrar ser més ràpids i més precisos en comparació amb tècniques tradicionals. També es va demostrar que els mètodes d'aprenentatge profund són més robustos davant la presència de soroll en la placa.

Palabras claves: *C. elegans*; Lifespan; Healthspan; procesaiento d'imatges; tècniques tradicionals; aprenentatge profund; detecció; esqueletizació, seguidor múltiple; imatges de baixa resolució; simulador de comportaments; imatges sintètiques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This Thesis project is part of the "RETOS INVESTIGACIÓN" CONOCE project (Ref. RTI2018-094312-B-I00, 2019-2021), entitled "Advanced behavioural monitoring of *Caenorhabditis elegans*, based on active vision, to analyze the cognitive function and aging" funded by "Ministerio de Ciencia, Innovación y Universidades", where all the advances achieved have been applied and validated.

At present, there are different types of organic models, mammalian, non-mammalian and even small-scale organisms, including bacteria. *Caenorhabditis elegans* (*C. elegans*) is one of these models, and one of the most widely used by laboratories in the study of diseases, pathologies, and the development of new drugs. They are about 1[mm] long, which makes them easy to handle and culture. They can be stored in large quantities in incubators (more than 1,000 per Petri dish). They are fed with bacterial strains such as *Escherichia coli* (*E. coli*), making their cultivation simple and economical. It is also transparent, making it easy to monitor internally in vivo. Its entire genome is mapped, and it has a 40% similarity to the human genome. These characteristics, as well as its lifespan of 15 to 30 days, make it a very attractive model for biological assays and experiments.

There are different strains or mutations of C. elegans, the best known are: Parkinson's disease (PD), Alzheimer's disease (AD), Huntington's disease (HD), and amyotrophic lateral sclerosis (ALS). The most common assays with this nematode are related to extending its life expectancy (lifespan) and improving its life conditions (healthspan). To perform both experiments, laboratory personnel follow their postures in Petri dishes to analyze their behaviour towards different reagents. This task involves an enormous amount of time and has been performed manually for a long time.

Nowadays, thanks to technological advances and digital image processing techniques (computer vision), the paradigm of repetitive and manual processes is being replaced by more automatic solutions. In the case of posture-tracking, much work has been done in the last decade, but this task presents a major challenge when the *C. elegans* aggregate with each other or with elements in their environment (noise). The flexibility of their bodies allows them to assume infinite postures, which makes behavioural analysis difficult. Many trackers choose to discard these cases of aggregation because of the difficulty of resolving this. However, preserving the identity of each individual (posture-tracking) even in cases of aggregation would allow us to take advantage of all the data from the study of the behaviours performed as well as to have measures of the interactions between *C. elegans*.

The most common aggregation cases are parallel aggregations, crossover, head/tail contact, or aggregations with Petri dish noise. To solve aggregation cases, some methods propose to use the skeleton because it contains the same pose information as the whole body. The resulting skeleton during an aggregation has branches which are analyzed to separate poses, but in cases of parallel aggregation or coiling with aggregations, these branches are not available. The first paper in section 1, chapter 2, presents a solution to this problem, for this, an improved skeleton is generated by creating these missing bifurcations using the width information of each *C. elegans* respectively. The results using this improved skeleton were shown to be more accurate compared to a traditional skeletonization method with statistically significant differences (see Section A.1).

The improved skeleton helped to obtain better results in posture tracking of *C. elegans*, however, preserving the identity of each individual in image sequences depends not only on the skeleton but also on other criteria. Identity loss or changes in image sequences are caused by a bad selection of criteria to solve optimization and also by the complexity of aggregation. Other work has shown that including criteria such as skeleton smoothness, or overlap with the previous pose influences these optimization algorithms, but even so, solving aggregations in low-resolution image sequences is a difficult task even for a human. The second paper in section 2, chapter 2, presents two new criteria, color, and completeness, which were used with the previous pose overlap criterion, and the improved skeleton were key to obtaining optimal posture-tracking in low-resolution image sequences.

Before starting the pose-tracking of *C. elegans*, other tasks such as filtering and detection are performed. These tasks are computationally expensive, but they allow us to recognize *C. elegans* from other objects (noise), making the pose tracking apply to our objects of interest. The first paper in section 1, chapter 2, presented a filtering and detection method that consisted of analyzing the image sequence to detect moving objects and filtering out static objects and objects with skeleton lengths different from *C. elegans*. This technique was able to filter out a lot of noise, however, objects with similar shapes and lengths to *C. elegans* were considered objects of interest. On the other hand, the third paper in section 3, chapter 2, proposed to detect postures of *C. elegans* and better filter out the noise using deep learning techniques. This paper compared two detection neural network architectures and showed that the smaller version of YOLOv5 is faster than the Faster R-CNN neural network.

Using deep learning techniques is now a very common practice. These techniques are often faster and more accurate than traditional techniques. But on the other hand, obtaining a specific dataset to train these deep learning models takes a lot of labelling time. That was the case for the third paper, the dataset used was very large and required many hours of labelling. To train more complex models, for example, to solve aggregation cases for postures-tracking of *C. elegans*, a dataset with many aggregation cases is needed. The postures that these *C. elegans* can have individually or in aggregation cases are very complex, making the labelling task almost impossible to perform. The fourth and fifth papers, sections 4 and 5, chapter 2, propose to solve this problem using a synthetic image simulator. This simulator randomly generated individual, rolling, parallel aggregation, and crossover aggregation behaviours, allowing to have a specific dataset to train two deep learning models. The first one [43], was a skeletonization model of the head, tail, and body of *C. elegans* to obtain its postures in cases of coiling and aggregation. The second (section 5) was a model for predicting the next poses to solve aggregation cases in image sequences. It is worth mentioning that these models were trained with purely synthetic images, but successfully tested using real images. The next pose prediction model was used to filter the solutions in paper 2, which allowed to improve the processing time of the optimizer and to obtain better solutions.

1.1 Objectives

One of the objectives of this thesis is the analysis of the state of the art of detection techniques, posture-tracking, and behavioural recognition of *Caenorhabditis elegans* (*C. elegans*). For this purpose, a study and comparison of techniques proposed by other authors will be carried out.

Achieved this search, the advantages and disadvantages of these techniques will be analyzed in order to develop new automatic methods that facilitate the different tasks of postures-tracking of the *C. elegans* in low-resolution image sequences.

Another objective is the design, development, and evaluation of advanced posture-tracking techniques for *C. elegans*. This will be performed using traditional computer vision and deep learning techniques.

Finally, a comparison will be made between traditional computer vision techniques and deep learning techniques in the detection and tracking of *C. elegans* postures. Experiments will be designed and developed using both techniques.

1.2 Contributions

- An improved skeletonization algorithm capable of finding better positions of *C. elegans* in cases of aggregation and coiling developed with traditional imaging techniques for low-resolution images. This method was also proposed:
  - A method for detecting *C. elegans* and filtering noise using traditional image processing techniques.
- A multi-tracker method for *C. elegans* postures based on the improved skeletonization algorithm to solve aggregation cases developed with traditional imaging techniques for low-resolution images. This multi-tracker method also proposed:
  - A multi-criteria optimization method to obtain the best posture of *C. elegans* in cases of aggregation and coiling.
- Two deep learning models, YOLOv5 and Faster R-CNN, trained to detect *C. elegans* in low-resolution images.
- A new synthetic generator of synthetic images of *C. elegans* behaviours in low-resolution images for training deep learning models.
- A multi-posture-tracking method of *C. elegans* to solve cases of aggregation in low-resolution images using traditional image processing techniques and deep learning techniques. This method also proposed:
  - A deep learning model based on the U-Net architecture, able to predict the next postures of *C. elegans* in aggregation, trained only on synthetic images and successfully tested on real images.

1.3 Structure of the thesis

This thesis has been carried out in the format of a compendium of articles. Three articles published in indexed journals, one article published in an international conference, and one article accepted in the indexed journal: International Journal of Computer Vision (IJCV). Each section of chapter 2 corresponds to an article and contains the same information as the published article except for the bibliography and supplementary material. The bibliography chapter at the end of the thesis contains all the bibliographical references from chapter 2 without repetition. This was done to avoid duplication of references between chapters and sections. On the other hand, the chapter Appendix contains supplementary information of the published articles.

The thesis is structured in five chapters, an appendix, and a bibliography. The first chapter, Introduction, describes the problem, the objectives, the structure of the thesis, and related works. The second chapter corresponds to the articles. The third and fourth chapters correspond to the discussion and conclusions of all the results obtained.

1.4 Related works

There are four other co-authored papers related to detection, re-identification, and posture tracking techniques for *Caenorhabditis elegans*. These healthspan and lifespan papers were also developed on low-resolution Petri dish images. Participation in these papers allowed the exploration of other automatic alternatives related to the subject area of this thesis and helped to improve the methods proposed in the development of this thesis. These papers are not part of the thesis, but are detailed in this section:

- The first paper, *"Reducing results variance in lifespan machines: an analysis of the influence of vibrotaxis on wild-type Caenorhabditis elegans"*, proposed a robust lifespan method capable of recognizing moving and standing behaviours and better detecting the postures of live and dead *C. elegans*. This method was designed with traditional computer vision techniques and allowed to reduce the error in the lifespan curve:

  "Nowadays, various artificial vision-based machines automate the lifespan assays of C. elegans. These automated machines present wider variability in results than manual assays because in the latter worms can be poked one by one to determine whether they are alive or not. Lifespan machines normally use a "dead or alive criterion" based on nematode position or pose changes, without poking worms. However, worms barely move on their last days of life, even though they are still alive. Therefore, a long monitoring period is necessary to observe motility in order to guarantee worms are actually dead, or a stimulus to prompt worm movement is required to reduce the lifespan variability measure. Here, a new automated vibrotaxis-based method for lifespan machines is proposed as a solution to prompt a motion response in all worms cultured on standard Petri plates in order to better distinguish between live and dead individuals. This simple automated method allows the stimulation of all animals through the whole plate at the same time and intensity, increasing the experiment throughput. The experimental results exhibited improved live-worm detection using this method, and most live nematodes (>93%) reacted to the vibration stimulus. This method increased machine sensitivity by decreasing results variance by approximately one

half (from ± 1 individual error per plate to ± 0.6) and error in lifespan curve was reduced as well (from 2.6% to 1.2%)."

Puchalt, J. C., Layana Castro, P. E., and Sánchez-Salmerón, A. J. (2020). Reducing results variance in lifespan machines: an analysis of the influence of vibrotaxis on wild-type *Caenorhabditis elegans* for the death criterion. Sensors, 20(21), 5981. https://doi.org/10.3390/s20215981.

- The second paper, *"Towards lifespan automation for Caenorhabditis elegans based on deep learning: analysing convolutional and recurrent neural networks for dead or live classification"*, proposed an improvement in the automatic generation of *C. elegans* lifespan curves by combining traditional computer vision techniques with deep learning techniques. The classifier trained with image sequences of live/dead *C. elegans* behaviours allowed better differentiation between these two cases and better results than traditional techniques. This paper also proposed a data augmentation technique based on synthetic image generation to train neural networks with small/medium datasets:

  The automation of lifespan assays with C. elegans in standard Petri dishes is a challenging problem because there are several problems hindering detection such as occlusions at the plate edges, dirt accumulation, and worm aggregations. Moreover, determining whether a worm is alive or dead can be complex as they barely move during the last few days of their lives. This paper proposes a method combining traditional computer vision techniques with a live/dead C. elegans classifier based on convolutional and recurrent neural networks from low-resolution image sequences. In addition to proposing a new method to automate lifespan, the use of data augmentation techniques is proposed to train the network in the absence of large numbers of samples. The proposed method achieved small error rates (3.54% ± 1.30% per plate) with respect to the manual curve, demonstrating its feasibility.

  García Garví, A., Puchalt, J. C., Layana Castro, P. E., Navarro Moya, F., and Sánchez-Salmerón, A. J. (2021). Towards lifespan automation for *Caenorhabditis elegans* based on deep learning: analysing convolutional and recurrent neural networks for dead or live classification. Sensors, 21(14), 4943. https://doi.org/10.3390/s21144943.

- The third article, *"A new training strategy for spatial transform networks (STN's)"*, proposed a new training method for STN neural networks to obtain better convergence. For this, a synthetic image simulator was developed to generate a dataset to average and analyze specific cases:

  Spatial transform networks (STN) are widely used since they can transform images captured from different viewpoints to obtain an objective image. These networks use an image captured from any viewpoint as input and the desired image as a label. Usually, these images are segmented, but this could lead to convergence problems if the percentage of overlap between the segmented images is quite low. In this paper, we propose a new training method to facilitate the convergence of a STN in these cases, even when there is no overlap between the object's projections in the two images. This new strategy is based on the incorporation of the distance transformation images to the training, thus increasing the useful image information to provide gradients in the loss function. This new training strategy has been applied to a real case, with images of Caenorhabditis elegans, and to a simulated case, which uses *artificial images to ensure that there is no overlap between the images used for the assays. In the assays carried out with these datasets, we have shown that the training convergence is strengthened, reaching a precision level for IoU metric of 0.862 and 0.984, respectively, and the computational cost has been maintained compared to the assay with segmented images, for the real case.*

Navarro Moya, F., Puchalt, J. C., Layana Castro, P. E., García Garví, A., and Sánchez-Salmerón, A. J. (2022). A new training strategy for spatial transform networks (STN's). Neural Computing and Applications, 34(12), 10081-10092. https://doi.org/10.1007/s00521-022-06993-0.

- Finally, the fourth paper, *"Analysis of a C. elegans lifespan prediction method based on a bimodal neural network and uncertainty estimation"*, proposed a new method for predicting lifespan curves using deep learning techniques. This bimodal neural network model was trained with purely synthetic data and tested with real images. Synthetic images were generated using different live/dead behaviours of *C. elegans* in Petri dish images.

  In recent decades, assays with the nematode Caenorhabditis elegans (C. elegans) have enabled great advances to be made in research on aging. However, performing these assays manually is a laborious task. To solve this problem, numerous C. elegans assay automation techniques are being developed to increase throughput and accuracy. In this paper, a method for predicting the lifespan of C. elegans nematodes using a bimodal neural network is proposed and analyzed. Specifically, the model uses the sequence of images and the count of live C. elegans up to the current day to predict the lifespan curve termination. This network has been trained using a simulator to avoid the labeling costs of training such a model. In addition, a method for estimating the uncertainty of the model predictions has been proposed. Using this uncertainty, a criterion has been analyzed to decide at what point the assay could be halted and the user could rely on the model's predictions. The method has been analyzed and validated using real experiments. The results show that uncertainty is reduced from the mean lifespan and that most of the predictions obtained do not present statistically significant differences with respect to the curves obtained manually.

  García-Garví, A., Layana-Castro, P. E., and Sánchez-Salmerón, A. J. (2023). Analysis of a *C. elegans* lifespan prediction method based on a bimodal neural network and uncertainty estimation. Computational and Structural Biotechnology Journal, 21, 655-664. https://doi.org/10.1016/j.csbj.2022.12.033.

Chapter 2

Papers

2.1 Paper1

Improving skeleton algorithm for helping *Caenorhabditis elegans* trackers

Pablo E. Layana Castro[1], **Joan Carles Puchalt**[1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Layana Castro, P. E., Puchalt, J. C., and Sánchez-Salmerón, A. J. (2020). Improving skeleton algorithm for helping *Caenorhabditis elegans* trackers. Scientific Reports, 10(1), 22247. https://doi.org/10.1038/s41598-020-79430-8.

2.1.1 Abstract

One of the main problems when monitoring *Caenorhabditis elegans* nematodes (*C. elegans*) is tracking their poses by automatic computer vision systems. This is a challenge given the marked flexibility that their bodies present and the different poses that can be performed during their behaviour individually, which become even more complicated when worms aggregate with others while moving. This work proposes a simple solution by combining some computer vision techniques to help to determine certain worm poses and to identify each one during aggregation or in coiled shapes. This new method is based on the distance transformation function to obtain better worm skeletons. Experiments were performed with 205 plates, each with 10, 15, 30, 60 or 100 worms, which totals 100,000 worm poses approximately. A comparison of the proposed method was made to a classic skeletonisation method to find that 2196 problematic poses had improved by between 22% and 1% on average in the pose predictions of each worm.

2.1.2 Introduction

C. elegans is a very important free-living transparent nematode that has been used since 1960 as a development model in the biology field. Monitoring these worms allows behavioural data to be acquired in their life cycles, especially at advanced adult age when they present some disorders, physical deterioration, tissue deterioration and mobility,

among other characteristics, similar to humans at advanced ages[92, 37]. This makes it an ideal model organism for studying and treating several types of diseases [57].
While investigating *C. elegans* behaviour, some advanced tools and techniques were developed to recognise the outward configurations, poses and coiled shapes that these nematodes can take when they roll up or aggregate with others while moving. Many of these systems require one end or both ends of *C. elegans* to be visible in order to identify and estimate a pose with different parameters, such as length [101], smoothness [77, 76], previous segmentation [95], or other complex models [56, 28, 36, 100, 19, 80].

Pose estimation models have been developed using neural networks, such as [25, 8, 47], which have been shown to have made accurate predictions in their datasets. These models solve the problems of finding individual worm poses, such as coiled shapes or even self-occluded, although these tools pose a formidable challenge because the learned network is difficult to interpret. This involves resorting to highly trained people to understand how this black-box works, and to even explain the decisions made or how the answer came about, because this information is hidden, unlike classic computer vision algorithms [101, 77, 76, 64] in which the result easily interpretable in each computational pipeline phase. Furthermore, neuronal networks have millions of parameters which are usually learned from annotated data, but annotated data is often very expensive to obtain. This problem can become intractable, since a large amount of annotated data is usually required to ensure model generalization.

The method herein proposed attempts to offer a simple solution to many of these complex coiled worm, self-occluded worm, and even fully aggregated, worm problems by means of simple computer vision techniques. This method improves the worm skeleton obtained by considering the width of each *C. elegans*. This new skeleton, unlike the classic one, allows us to divide highly connected bodies which provide information for the pose tracking problem.

2.1.3 Methods

C. elegans strain and culture conditions

The studied *C. elegans* are wild-type (N2) and CB1370 *daf-2* (e1370), obtained from the Caenorhabditis Genetics Center at the University of Minnesota. Experiments were performed at 20ºC in nematode growth medium (NGM) seeded with *Escherichia coli* strain OP50 (*E. coli*) as standard diet. Synchronised young adult *C. elegans* were obtained from worm eggs incubated at 20ºC in 55 mm-diameter NGM plates. FUdR (0.2 mM) was used to prevent reproduction. All the plates were closed with a lid and fungizone (1 µg/ mL) was added to prevent fungal contamination [88]. Plates were cultivated with 10, 15, 30, 60 and 100 worms to obtain a wide variability of cases, aggregation of two *C. elegans* or more, and single rare poses (e.g. coiled shapes, curls, etc.).

Capture conditions for image sequences

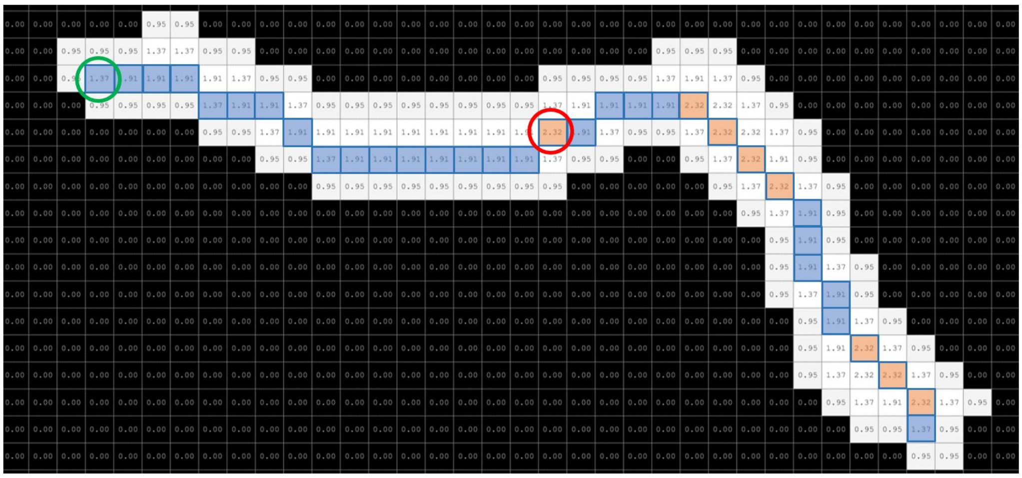
A laboratory operator collects plates from the incubator and places them in the image capture system [64], where a sequence of 30 images is captured at a frequency of 1Hz. The time outside the incubator is short, and the ambient temperature is 20ºC to avoid condensation. If condensation appears on plate covers, they are removed by the operator before the image acquisition process. All data was collected with lids, since no condensation problem was detected. *E. coli* OP50 grass was seeded in the centre of plates to prevent worms from going outside the field of view by scaling the edges of plates or near them.

Illumination of the image capture system

Artificial vision systems depend on various components that work together to acquire, process and analyse images. The lighting system is one of the most important as it allows notable characteristics to be highlighted on images which were, in our case, *C. elegans* processing and the background. The employed technique was an intelligent active backlight illumination that consisted of putting the Petri plate between the lighting system and the camera. As proved in [64], this lighting method allows constant intensity levels of the background (near the control reference of 48) and the studied *C. elegans* (near the 0 intensity level) to be obtained, which means that it is more robust than standard backlight methods as it narrows the variability of the captured images, and allows fixed segmentation thresholds to be used for all images.

Image acquisition method

The image sequence was acquired at a resolution of 1944x1944 pixels and a frequency of 1Hz (1 image per second) using the system that uses an RGB Raspberry Pi v1.3 camera, OmniVision OV5647, 2592 × 1944 pixels resolution, 1.4 × 1.4 µm pixel size, 53.50° × 41.41° field of view, with the 1/4 ″ optical size and 2.9 focal length, a lighting system based on a 7 Raspberry Pi display at a resolution of 800 × 480 and 60 fps, 24-bit RGB colour and a Raspberry Pi 3 processing unit. This developed image acquisition system is open hardware. Its assembly procedure, parts and description are detailed in another work [64]. Under these image acquisition conditions, a young adult worm or an adult worm on the plate is projected on image Figure 2.1, with a size of 20 or 40 pixels of length and 3 or 5 pixels of width.



0.00	0.00	0.00	0.00	0.00	0.95	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.95	0.95	0.95	1.37	1.37	0.95	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.95	1.37	1.91	1.91	1.91	1.91	1.37	0.95	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	0.95	0.95	1.37	1.91	1.37	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.95	0.95	0.95	0.95	1.37	1.91	1.91	1.37	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	1.37	1.91	1.91	1.91	1.91	2.32	2.32	1.37	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	1.37	1.91	1.91	1.91	1.91	1.91	1.91	1.91	1.91	1.91	2.32	1.91	1.37	0.95	0.95	1.37	2.32	2.32	1.37	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	1.37	1.91	1.91	1.91	1.91	1.91	1.91	1.91	1.37	0.95	0.95	0.00	0.00	0.95	1.37	2.32	1.91	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.00	0.00	0.00	0.00	0.00	0.95	1.37	2.32	1.37	0.95	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.37	1.91	0.95	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.91	0.95	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.91	1.37	0.95	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.37	1.91	0.95	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.91	1.37	0.95	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.91	2.32	1.37	0.95	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.37	2.32	2.32	1.37	0.95	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	1.37	1.91	2.32	1.37	0.95	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	1.37	0.95	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95	0.00	0.00

Figure 2.1: **Worm model.** Skeleton (blue and orange pixels). Values are the result of the distance transform; Worm model: maximum width value (red circle): 2.32 x 2 pixels; minimum width value (green circle): 1.37 x 2 pixels; total length value: 34 pixels.

Distance transform function

The distance transform or Euclidean distance map is a tool that transforms an input binary image into a grey-scale image, where each transformed pixel is a decimal value or a brightness value (the values of each pixel in Figure 2.1) which corresponds to the Euclidean distance of that pixel to the closest background pixel [81].

Classical skeletonisation method

Many classic skeletonisation methods are available as computer vision tools, and each one obtains different skeletons results. We decided to use the Matlab *bwmorph* function as an example of the classic image skeletonisation method. This function's input parameters include: first, the binary image to skeletonise; second, the operation to be performed on the binary image, in our case 'thin'; third, the value number of times to apply the operation. If this number is 'Inf', the operation will be repeated until the result remains unchanged (coloured pixels in Figure 2.1).

Worm model

The worm model is defined by the following three parameters: the maximum and minimum width values and total length value, obtained by the former methods. To estimate the maximum and minimum width value (maximum value in the red circle in Figure 2.1), firstly the transform distance values of the worm image segmentation are obtained. Then the skeleton is obtained with the classic skeletonisation function. On this skeleton, the pixels with the highest and lowest value are found. Total length is the total number of pixels in the skeleton.

Processing image sequences

As previously mentioned, *C. elegans* are ideal organic models for studying and treating age-related diseases. The most popular assays are lifespan and healthspan. As these assays are usually performed with several synchronised worms per plate, it is important to monitor their behaviour individually or when they aggregate with others. There are some multitrackers available for *C. elegans* [19, 89, 74, 33, 17, 70], but they require manually adjusting some parameters like the segmentation threshold. In our case, we used a simple and fully automatic image processing pipeline to take advantage of the intelligent illumination system [66]. The main processing steps to resolve the tracking of worms throughout the imaging sequence are outlined below.

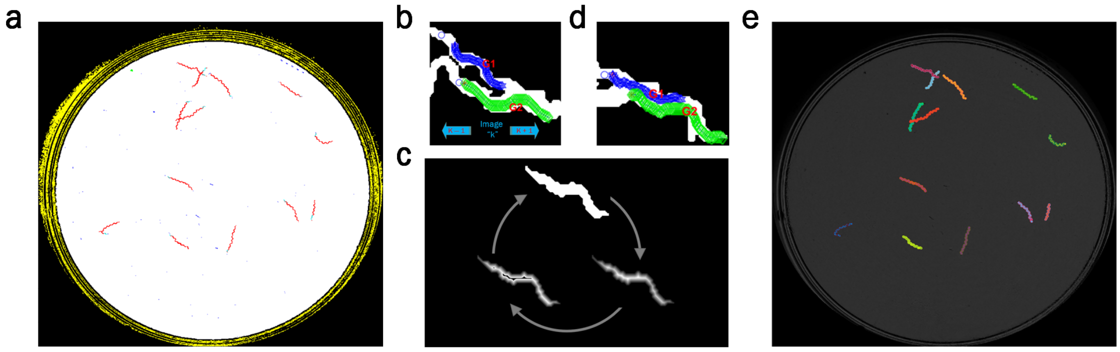


Figure 2.2: **Image processing pipeline.** (**a**) Image sequence segmentation result (background in black, region of interest in white, worm tracks in red, noise in blue). (**b**) Worms before aggregation. (**c**) Proposed skeleton technique to obtain separate skeletons. (**d**) Optimisation results (predicted poses). (**e**) *C. elegans* tracking results.

Firstly, the region of interest on the Petri plate is obtained from each image (white circle in Figure 2.2a) by applying a simple threshold as all the pixels with an intensity value above 35 on the grey scale belongs to this region of interest. It is worth remarking that work can be done with a fixed segmentation threshold for all these images because the image capture system incorporates a lighting control strategy to keep background levels constant [64]. After this segmentation, an AND operation between all the segmented images in a sequence is run. A close operation (dilating and eroding) is applied to the resulting image to fill in small holes. Finally, the largest connected component is detected as the region of interest.

In parallel to this first step, we also segmented the tracks of *C. elegans* (see Figure 2.2a). This process consists of attempting different threshold levels and analysing whether the skeleton of each segmentation corresponds to a minimum expected length. Otherwise it is classified as noise (blue).

Each track is analysed in the second step. For these tracks, the worms in each image are segmented. If it is a multiworm track, images in the sequence are searched for, where all the *C. elegans* are separate and not self-occluded (Figure 2.2b). These simple cases allowed us to estimate the worm model, defined by length and width features (Figure 2.1). The tracking algorithm starts from one of these simple cases to obtain the worm model before analysing all the sequence both forwardly and backwardly. A new skeleton technique, described in the next section, is proposed to deal with aggregation or self-occluded worms by considering the estimated widths of *C. elegans* (Figure 2.2c). Afterwards, an optimisation algorithm is used to obtain the best skeletons (Figure 2.2d). Finally, Figure 2.2e shows all the tracked worms on a plate.

Proposed skeletonising method

Classic image skeletonisation methods, used in computer vision, present major errors in some cases, and may even cause loss of worm identity during tracking. Specifically, these errors occur in those cases in which *C. elegans* are self-occluded or fully rolled (Figure 2.3a)), when they collide with noise on plates (Figure 2.3b)), come the head, tail or part of the body comes partially into contact with another worm (Figure 2.3c)), when two worms travel in parallel for several instants of time (Figure 2.3d)), etc. The proposed skeleton method obtains a better skeleton in the former cases (Figure 2.3i-j-k-l) compared to classic methods (Figure 2.3e-f-g-h). New skeletons are more exact to the centreline of the current position of *C. elegans*, which helps to solve the tracking problem. *C. elegans* bodies can be reconstructed from skeletons by a dilate operation (Figure 2.3m-n-o-p).

The proposed method consists of the following steps. Firstly, a subimage of the bounding box containing the segmented connected region (or blob) is cropped from the current image (Figure 2.4a). A blob can be composed of one isolated worm, some connected worms, or others.

Secondly, the sub-image is scaled by a factor of 3 (Figure 2.4b). Scaling is optional and will be applied, when low-resolution images are used, to more precisely separate aggregated bodies. Thirdly, a distance transform is applied to the connected region (Figure 2.4c).

Fourthly, some blob pixels are transformed from the foreground to the background. This transform is known as background transform. The main idea of this method is to transform the pixels in the background to separate connected areas when the blob width is wider than the expected worm width. This means that if there are pixels in the blob with a distance value higher than the maximum width of each model, then these pixels change their values to zero (background value) (Figure 2.4d). This process is repeated iteratively until no pixel transforms in the background.

With aggregation, the used width value is the maximum width value estimated in the worm model. If the worm model could not be estimated before aggregation, a default width value can be used. This default value is easy to define because worms are synchronised. This means that all the worms in an experiment are of comparable size (length and width).

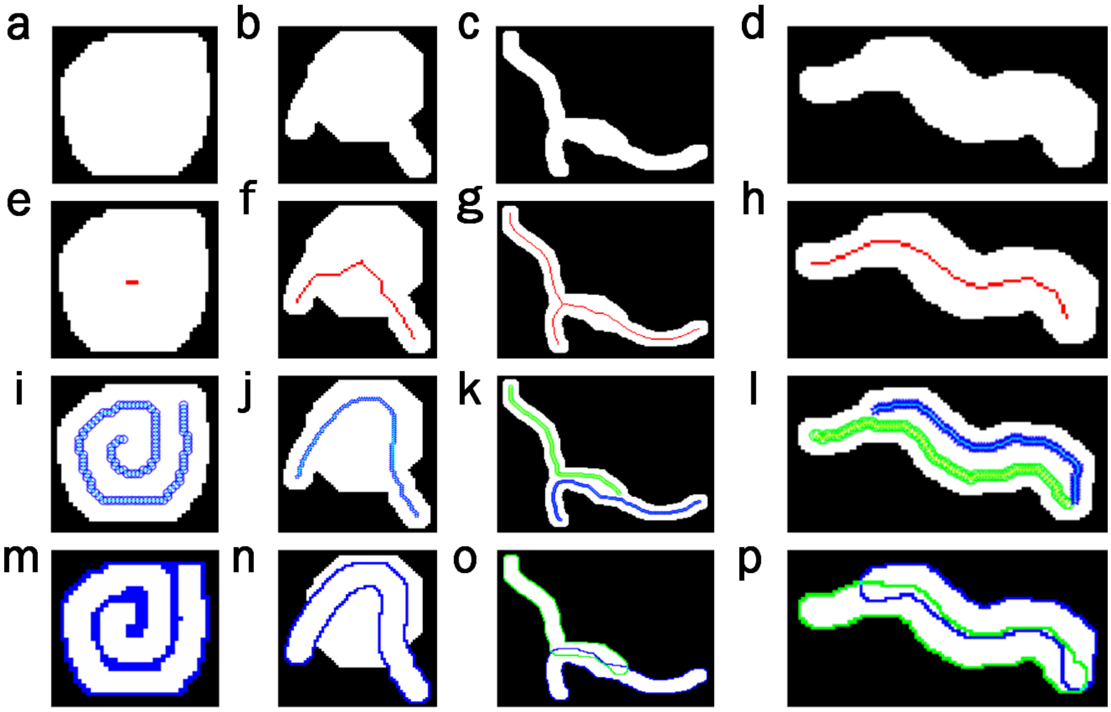


Figure 2.3: **Comparison of skeleton methods.** Each of *C. elegans* columns fully wound, *C. elegans* colliding with noise on the plate, *C. elegans* partially in contact, two *C. elegans* traveling in parallel, which correspond to different cases that present problems when tracking is done by a classic skeletisation method with possible loss of identity. (**a,b,c,d**) Worm segmentation. (**e,f,g,h**) Skeletons obtained by a classic computer vision method in red. (**i,j,k,l**) Skeletons obtained by the new proposed method in blue and green. (**m,n,o,p**) Reconstruction of the *C. elegans* bodies from the skeletons obtained by the proposed method.

Fifthly, a classic skeleton method is applied to the modified blob (after applying the background transform) to obtain separate skeletons (Figure 2.4e). It is worth remarking that the proposed method returns the same skeleton as the classic method when no blob pixel is background transformed. In this case, all the distance transform values fall within the expected width.

Finally, to obtain the best solution (Figure 2.4f), all the combinations of poses are analysed (Figure 2.5) and the minimum cost is calculated with an optimisation algorithm that evaluates length, width characteristics and the earlier pose state, as described in [66]. All the possible poses are obtained by exploring the continuous paths connecting endpoints and intersections, constrained to the expected lengths. If there are no endpoints and intersections, all the skeleton points overlapping earlier poses are taken. The resulting combinations of poses are filtered to remove similar poses for reducing computational cost. Poses are considered similar when they overlap at least 90 percent.

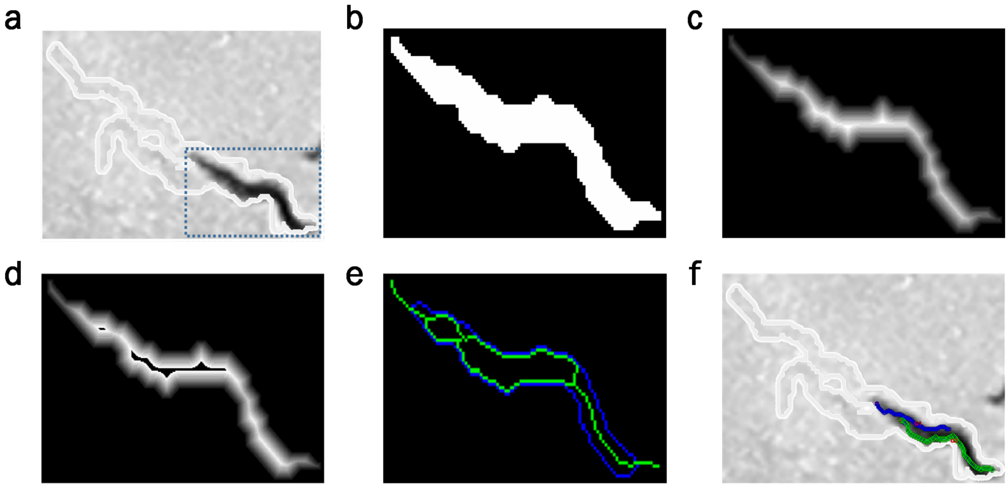


Figure 2.4: **Example of tracks of two connected *C. elegans*.** Proposed method steps. (**a**) Cropped subimage containing a current blob. (**b**) Segmentation of the current subimage. (**c**) Distance transform coded by grey intensity levels. (**d**) Background transform by using the maximum width value. (**e**) New skeletons. (**f**) Optimisation result (predicted poses).

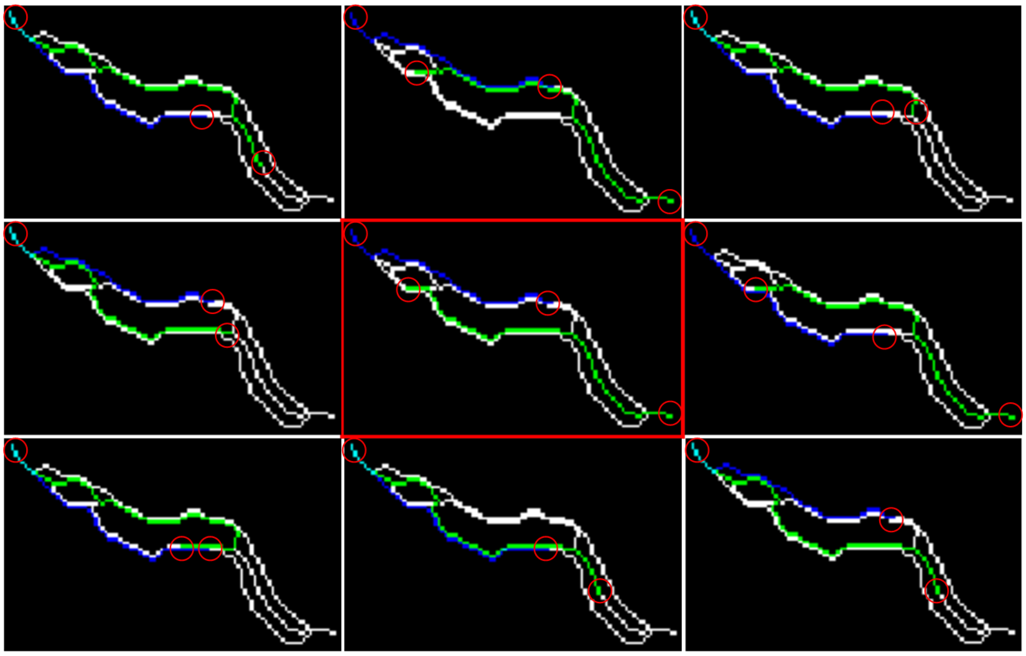


Figure 2.5: **Possible skeleton combinations.** Skeleton combinations of two connected worms (blue and green) that will be evaluated by an optimiser to obtain the best possible solution marked by the red square.

Considerations for rolled *C. elegans* in the proposed skeletonising method

If a *C. elegans* crawls individually, we proceed similarly with the former method, but the width values to be used in the background transform are the minimum ones in our worm model in order to consider all the coiled shapes, and cases such as coiling completely or partially (Figure 2.6a-b-c-d-e-f-g-h). In this example, a second background transform iteration is needed (Figure 2.6e). If holes have fewer than 4 pixels in the current resulting skeleton (fig6f), they are filled, and the skeleton is obtained again (Figure 2.6g). Filling holes avoids similar skeletons being obtained as possible solutions.

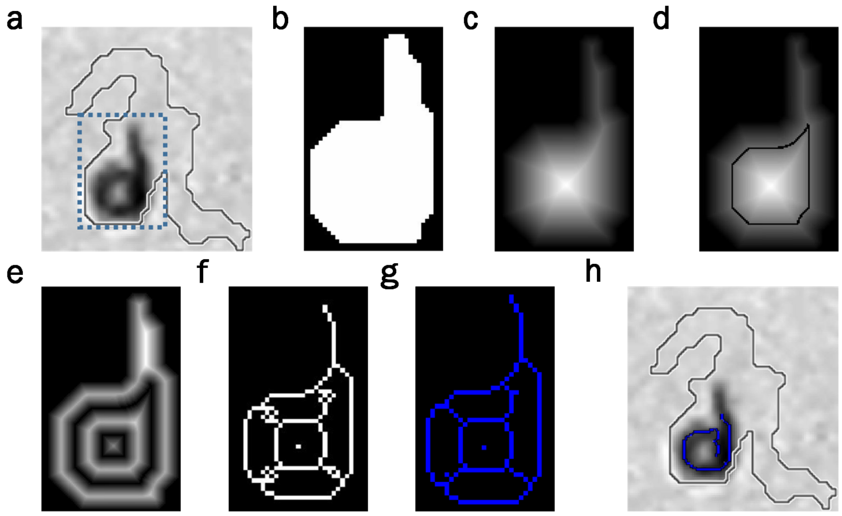


Figure 2.6: **Example of a track of a rolled worm.** Proposed method steps. (**a**) Subimage containing the current connected region. (**b**) Segmentation of the current subimage. (**c**) Distance transform. (**d**) Background transform by using the minimum width value. (**e**) Second iteration of the background transform. (**f**) Skeleton with holes. (**g**) Final skeleton. (**h**) Optimisation result.

Finally, to obtain the best solution (Figure 2.6h), all the possible solutions extracted from the resulting skeleton are analysed (Figure 2.7). The possible skeleton solutions are obtained by exploring any continuous path connecting the endpoints and intersections, constrained to the expected length.

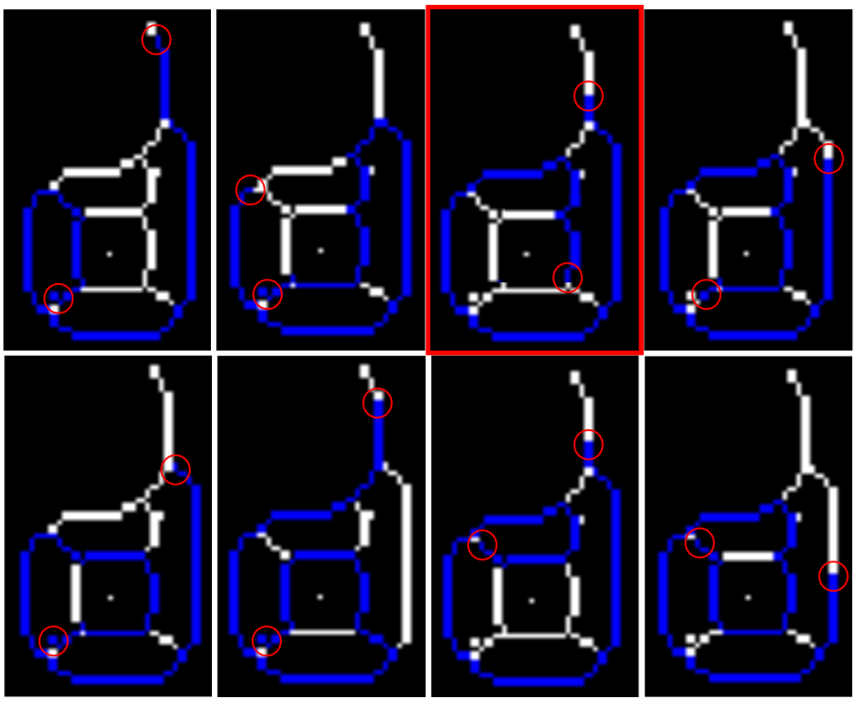


Figure 2.7: **Possible rolled skeleton combinations.** Possible rolled skeleton solutions (blue) using the proposed skeletonising method that will be evaluated by an optimiser to obtain the best possible solution marked by the red square.

Validation method

As a reference to compare the results obtained by the different skeletisation methods, ground-true manually labelled skeletons were annotated. This human annotation process was assisted by an application with a friendly interface, designed for helping to select every pixel belonging to the skeleton of each *C. elegans* one by one. This operation was performed for all 205 plates, including the 2196 poses used in the validation phase. The shape of these nematodes was recovered from the annotated skeletons by a dilation operation, applied to the annotated skeletons, with a disk shape of radius that equalled half the width (approx. 1 pixel or 2). This operation returned a shape of 3 or 5 pixels of width. Two pose overlapping indices were used to assess how the methods performed in the validation phase. Both indices were based on the Jaccard coefficient, or junction intersection (IoU). This coefficient measures the degree of precision of the location or detection of objects [75] and, as its name indicates, it is obtained by dividing the total area of the intersection of elements over the union of these areas [39] (Equation (2.1)). For this evaluation, we used the area of the reconstructed bodies from the manually labelled skeletons, the new skeletonising method and the classic skeleton method.

$$IoU = \frac{\sum P_{w1} \bigcap P_{w2}}{\sum P_{w1} \bigcup P_{w2}} \tag{2.1}$$

The first IoU index was expected to be higher because we compared a predicted pose (Figure 2.8b) to an annotated ground-true pose (Figure 2.8a) which must overlap (Figure 2.8c, d, e). The results for the shown example are IoU = 0.9784, 0.5667, and 0.2649, respectively. However, the second IoU index was expected to be lower because we compared two predicted poses, in the event of aggregation, which must no overlap one another: Figure 2.9a, b, c. The results for the shown example are IoU = 0.2297, 0.8049, 0.3387 for the reconstructed bodies from the manually labelled skeletons, the skeletons obtained with the classic method and the skeletons obtained by the proposed method, respectively.

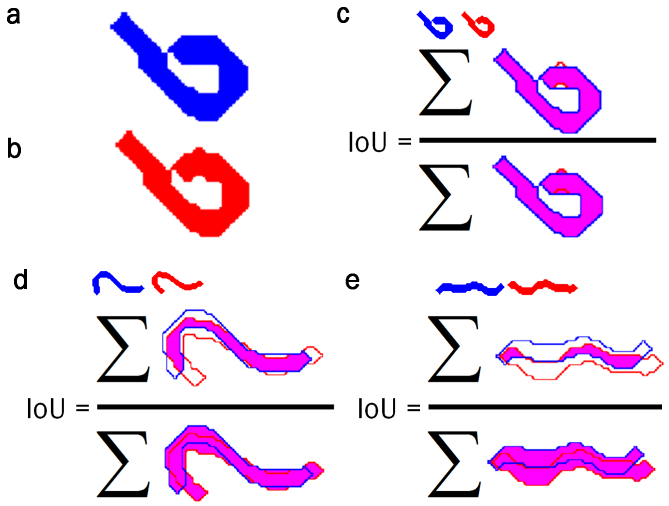


Figure 2.8: **First IoU index.** This index evaluates how close the response of the automatic method comes to the reference, and compares both obtained results to determine the improvement of one method compared to the other. The higher the IoU value, the closer the response comes to the reference. The evaluation is performed by reconstructing the skeletons obtained with a radio 2 disk. (**a**) Reconstructed body of the manually labelled skeleton. (**b**) Reconstructed body of the skeleton obtained from the new way to skeletonise. (**c, d, e**) Evaluation of the reconstructed skeletons.

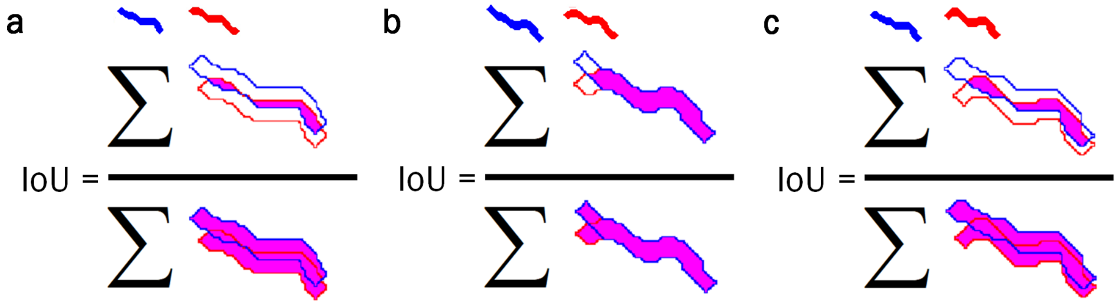


Figure 2.9: **Second IoU index.** This index measures how connected two resulting bodies are to one another. The lower the IoU value, the further apart bodies are. Like the earlier method, the evaluation is performed by reconstructing the skeletons obtained with a radio 2 disk. (**a**) Reconstructed bodies from the manually labelled skeletons. (**b**) Reconstructed bodies from the skeletons obtained with the classic method. (**c**) Reconstructed bodies from the skeletons obtained by the proposed method.

The IoU results obtained for each automatic method of 205 plates were analysed by the Wilcoxon Signed Ranks test and T-student statistics with the Statistics and Machine Learning Toolbox of Matlab2018b to obtain the comparison statistics between the two methods and to determine the degree of precision in relation to the reference and the reconstructed body of the manually labelled skeleton (see the complementary material).

Source code

The programme was developed in Matlab2018b in Windows 10 and works correctly in later versions with the image processing package. Its source code is in GitHub. It is open-source MIT (Massachusetts Institute of Technology) and can be downloaded from the repository at https://github.com/playanaC/Skeletonization.

2.1.4 Experiments and results

Experiments were performed with 205 plates, of which 193 corresponded to the plates with 10 and 15 worms, 1 plate with 30 worms, 3 plates with 60 worms and 8 plates with 100 worms, which totals 100,000 worm poses. All these data were analysed to obtain all kinds of problematic poses. As proven in [66] the bigger the number of worms per plate, the more likely them coming into contact with one another. Firstly, 205 plates were analysed, and all the tracks with poses where classic methods fail or have problems identifying each worm were selected in the validation dataset. This dataset contains 2196 poses (2.2% of all poses), including travelling in parallel, partial or total contacts, and coiled shapes. The percentage of contact poses agrees with the contact probabilities in [66, 89]. For the analysis, the above-described IoU index1 and 2 were used to compare the manually obtained reconstructed bodies of skeletons (reference), the skeletons obtained with the new method (new Skel) and the skeletons obtained using Matlab function *bwmorph* (classic Skel).

Index1 was employed to evaluate all the mean data for the New skel and classic skel samples. Two tests were used to analyse normality: the Kolmogorov Smirnov test for large samples (n>50) and the Shapiro Wilk test for small samples (n<50). The p-value was the criterion adopted to decide suspected normality, which was performed on the difference in both methods. If the p-value exceeded or equalled the level of significance (5%), the null hypothesis H0 was accepted (data came from normal distribution), otherwise alternative hypothesis H1 was accepted (data did not come from normal distribution). The data from normal distribution were analysed by the Student's t-test. The data that did not come from normal distribution were analysed by the Wilcoxon signed rank test.

The obtained results showed that our method significantly improved the pose prediction of each *C. elegans* versus the classic method, which allowed us to preserve identity in more cases and not lose the target during tracking. Table 2.1 shows the average and normalised results obtained for the different problematic cases, their respective deviations and the improvement achieved using the first IoU index.

			Mean IoU skel		Standard Deviation		Results %
Problematic cases	Total tracks	Total pose	New	Classical	New	Classical	Improvement
Self-occluded	212	803	0,77	0,76	0,14	0,13	0,91
Noise contact	17	509	0,68	0,66	0,22	0,23	2,07
Partial bodies aggregation	53	828	0,7	0,68	0,18	0,17	2,25
Full bodies aggregation	4	56	0,69	0,47	0,2	0,26	21,5

Table 2.1: **Summary of the comparison of automatic methods with the first IoU index.** This table includes the results obtained from the validation dataset and the improvements for each case using the first IoU index. The evaluated values show the percentage of success compared to the reconstructed body of the manually labelled skeleton (see the Supplementary Material for more details).

The results analysed by the Kolmogorov Smirnov test (N=2196) showed that the data did not come from normal distribution (p-value = 4.47E-143). The Wilcoxon Signed Ranks Test was used to analyse data (Figure 2.10), which revealed a statistically significant difference between both methods, with a p-value = 1.99E-18 less than the significance value of 0.05. The proposed method, New_skel, showed a significant improvement (mean = 0.7216, std. deviation = 0.1805, variance = 0.033) over the classic method, classical_skel (mean = 0.6996, std. deviation = 0.1839, variance = 0.034).

Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	1293	a	1092.0634	1412038
Negative	860	b	1054.3523	906743
Ties	43	c		
Total	2196			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

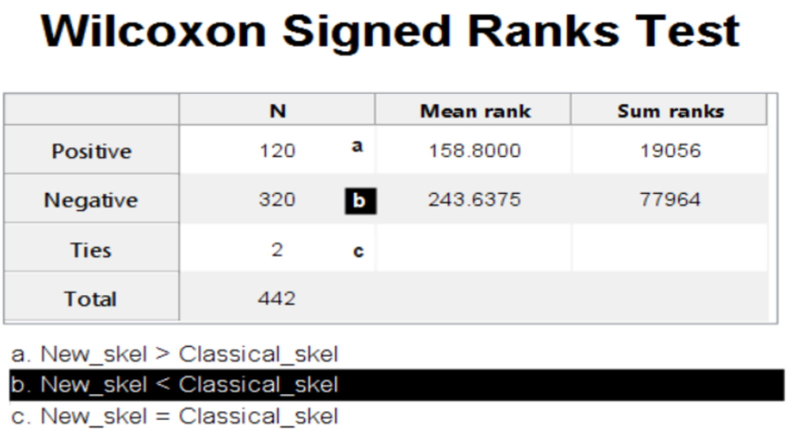
Figure 2.10: **Wilcoxon Signed Ranks Test for the mean-IoU data in Table 1.** The graph obtained from the Wilcoxon Signed Ranks test using the Statistics and Machine Learning Toolbox of Matlab2018b. Statistical toolbox indicated a higher positive range value than the negative range value. The p-value of this test was 1.998E-18 less than the significance value. We can thus conclude that a statistically significant difference appeared between both methods, and the proposed method (New_skel) achieved major improvement compared to the classic method (classical_skel).

Table 2.2 shows the average and normalised results obtained for connected bodies poses, their respective deviations and the improvement obtained using the second IoU index.

			Mean IoU skel		Standard Deviation		Results %
Problematic cases	Total plates	Total pose	New	Classical	New	Classical	Improvement
Partial bodies aggregation	53	414	0,07	0,1	0,08	0,11	3,83
Full bodies aggregation	4	28	0,21	0,69	0,2	0,15	48,33

Table 2.2: **Summary of the comparison of automatic methods with the second IoU index.** This table offers the results obtained from the connected bodies poses included in the validation dataset using the second IoU index for each problematic case. Values reveal the percentage of how connected they were to one another (See the Supplementary Material for more details).

Index2 was used to evaluate all the mean data for the New skel and Classical skel samples. The results analysed with the Kolmogorov Smirnov test (N=442) showed that the data did not come from normal distribution (p-value= 7.453E-59). The Wilcoxon Signed Ranks Test was used to analyse data (Figure 2.11) and indicated a statistically significant difference between both methods, with a p-value=2.56E-28 less than the significance value of 0.05. The proposed method, New_skel, indicated significant improvement (mean = 0.0754, std. deviation = 0.09658, variance = 0. 009) over the classic method, classical_skel (mean = 0.1419, std. deviation = 0.18256, variance = 0.033).



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	120	a	158.8000	19056
Negative	320	b	243.6375	77964
Ties	2	c		
Total	442			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

Figure 2.11: **The Wilcoxon Signed Ranks Test for the mean-IoU data in Table 2.** The graph obtained from the Wilcoxon Signed Ranks test using the Statistics and Machine Learning Toolbox of Matlab2018b. Statistical toolbox showed a negative ranges value that was higher than the positive ranges value. The p-value of this test was 2.56E-28 less than the significance value. So we can conclude that a statistically significant difference appeared between both methods and that the proposed method (New_skel) indicated a significant improvement over the classic method (classical_skel).

Finally, a sequence of two worms crawling in parallel is shown in Figure 2.12. In this case, the best solutions taken from classic skeletons completely overlap, with an undesired high value for the second IoU index. However, the best solutions obtained from the new skeletons overlap less, as expected, for these two worms.

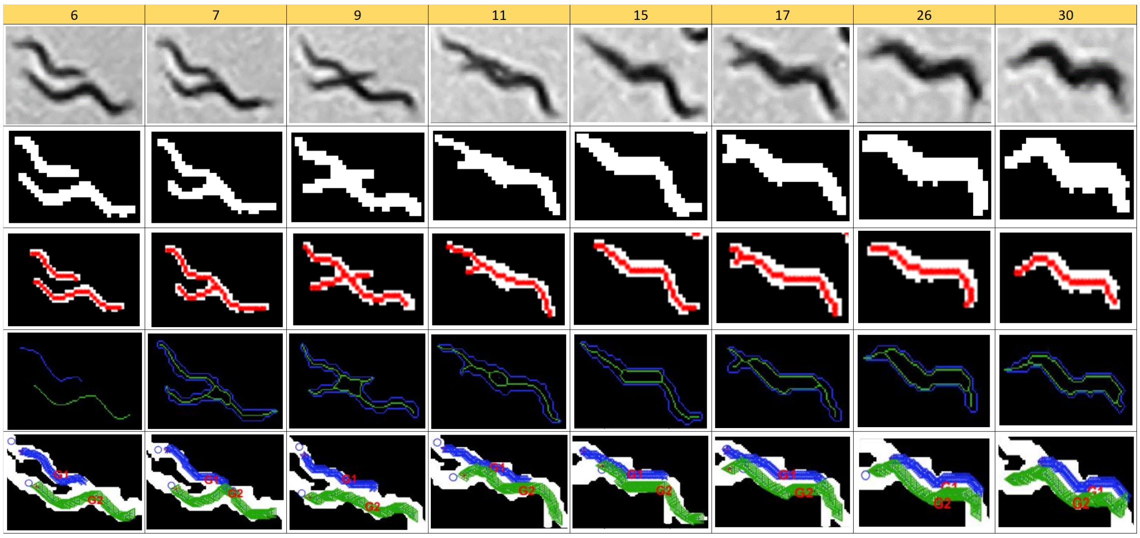


Figure 2.12: **Tracking sequence comparison with both automatic methods.** First row: original images; Second row: segmented images; Third row: Classic method skeletons; Fourth row: Proposed method skeletons; Fifth row: Best solutions from new skeletons.

2.1.5 Discussion

This work presents a new method to obtain better *C. elegans* skeletons to help tracking. This method uses earlier information about the estimated length and widths of each worm before aggregation, self-occlusion, or coiling to change the segmentation image. Skeletonising allows the complex behaviours of these worms to be analysed without losing information on the characteristics of their shape but can, in turn, present problems from reducing information, such as travelling in parallel or rolling up on themselves. All this leads classic methods to fail, and lose identity, location and poses during tracking, as a result of the very little information taken into account. Our method aims to provide a simple solution to this problem by obtaining a new skeleton with information on the width of each worm. This method consists of transforming pixels from the foreground to the background in segmentation whose distance-to-background values are higher than the estimated width. This allows the bodies of each *C. elegans* to be separated to obtain the best possible solution by solving the identity problem of each nematode. A priori, it seems interesting to apply the background transform only to a thin line of pixels. For example, when separating two worms crawling in parallel, it would be interesting to draw a thin background line which cuts the connections between both bodies. However, a problem occurs when attempting to separate these bodies with a thin line (only one pixel width). In this case, skeletons go through diagonal sections of this thin line as the classic skeleton works with 8-connected criteria to connect skeleton pixels. To avoid this problem, a scale factor of 3 is applied to increase image resolution, and then wider lines are used to avoid skeleton connections going through this line. In an individual worm case, such as an entire rolled body, drawing background lines with a width that equals the minimum width model (at least 3 scaled pixels) suffices to solve this problem. However with the aggregation of full bodies, another problem can occur because small discontinuities can appear in the middle of this line. These small holes allow hundreds of similar possible solutions, which incur a high cost to process them all. To solve this problem, we increased the line width to the maximum width model (approx. 9 scaled pixels) to fill these holes and to draw a continuous background line. This extra width reduces pose accuracy, but this reduction is negligible when going back to the low-resolution space. The proposed method returns the same skeleton as the classic method when no blob pixel is transformed from the foreground to the background. This means that all the distance transform function values fall within the expected width. This is the most probable case, where classic methods work well. With 10 or 15 worms per plate (55 mm-diameter), the probability of worm contact is 1%. This probability rises to 2.6%, 4.6%, and 6.6% with 30, 60, and 90 worms per plate, respectively, as proven in [66]. Therefore, classic methods are used with small worm numbers per plate (10, 15 or 30 worms), where the probability of errors is low. However, the proposed method can work with bigger worm numbers per plate and does not diminish tracking performance. Our experiments offer successful results, even with 100 worms per plate. Increasing the number of worms per plate is an interesting issue because it helps to reduce the cost of *C. elegans* experiments. The presented method can be adapted to any multitracker and can be implemented easily in other programming languages given the versatility of the distance transform function, which is found in many image processing toolkits.

2.2 Paper2

Caenorhabditis elegans multi-tracker based on a modified skeleton algorithm

Pablo E. Layana Castro[1], **Joan Carles Puchalt**[1], **Antonio García Garví** [1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Layana Castro, P. E., Puchalt, J. C., García Garví, A., Sánchez-Salmerón, A. J. (2021). *Caenorhabditis elegans* multi-tracker based on a modified skeleton algorithm. Sensors, 21(16), 5622. https://doi.org/10.3390/s21165622.

2.2.1 Abstract

Automatic tracking of *Caenorhabditis elegans* (*C. egans*) in standard Petri dishes is challenging due to high-resolution image requirements when fully monitoring a Petri dish, but mainly due to potential losses of individual worm identity caused by aggregation of worms, overlaps and body contact. To date, trackers only automate tests for individual worm behaviors, canceling data when body contact occurs. However, essays automating contact behaviors still require solutions to this problem. In this work, we propose a solution to this difficulty using computer vision techniques. On the one hand, a skeletonization method is applied to extract skeletons in overlap and contact situations. On the other hand, new optimization methods are proposed to solve the identity problem during these situations. Experiments were performed with 70 tracks and 3779 poses (skeletons) of *C. elegans*. Several cost functions with different criteria have been evaluated and the best results gave an accuracy 99.42% in overlapping with other worms and noise on the plate using the modified skeleton algorithm and 98.73% precision using the classical skeleton algorithm.

keyword: *C. elegans* assays; Lifespan; Healthspan; Image detection; Multi-tracker; Standard Petri dishes.

2.2.2 Introduction

The nematode *Caenorhabditis elegans* (*C. elegans*) is a widely studied animal model as its diverse age-related behavioral patterns provide valuable information on the function of its nervous system and is, therefore, an attractive model to evaluate the effects of mutations [57]. This facilitates the study and treatment of aging, as well as age-related pathologies and neurodegenerative disorders in humans at advanced ages [92, 37]. Many of these studies have shown that automatic tracking applications based on computer vision systems help to reduce the manual cost of data acquisition and research hours, improving potential observation of the effects of drug trials [86] and improvements in lifespan or “shelf-life” [66, 35, 65]. These systems provide quantitative information on alterations in the individual motility and behavior of worms produced by chemical substances in their environment (chemotaxis), providing statistical data that allow further research in the field of health and wellness.

C. elegans demonstrate group behavior [84, 12, 87], among the best known are courtship, mating, aggression, rearing and foraging. Group behavior assays are currently being performed, for example, research into the effect of 0_2 in food search analysis, and aggregation [78, 16]. These assays, like others, are visualized manually, due to the complexity of solving the identification problem during an overlapping or body contact of these worms. Currently, the automatic [74, 89, 70, 17, 83, 32, 6, 18, 50] or semi-automatic applications discard the data from tracks where there are these particular cases (overlapping and bodies contacts) [33, 38, 45]. Overlapping can take place among worms or may also be due to plate noise. Plate noise was defined as segmentation errors due to edges, or opaque waste in the plate.

Certain applications use diverse techniques and methods such as length [101], smoothness [77, 76], previous segmentation [100], or other complex methods [28, 19, 80, 56, 95, 36] to solve tracking problems during aggregation. The aforementioned methods were tested by our team and results indicated that previous segmentation is one of the most significant criteria to help identify worms within an aggregation. This was taken as a starting point to design different trackers, adding and combining numerous criteria, and it was found that the criteria of completeness and color used with the improvement of the form of skeletonizing [41] can help to identity the worms in the next image. Also, we introduced three new criteria: length, smoothness, and noise, revealing that they can be useful in re-identification.

We present different multi-trackers which, unlike any other trackers, are fully automatic (Table 2.3). Segmentation and identification of the edge, the interior rim of the Petri dish (area of interest), and segmentation tracks made by worms are performed without the intervention of human operators. These new methods show worm-tracking accuracy of above 98% in nematode aggregations and problems related to plate noise. The best result reaching 99.42% accuracy.

Table 2.3: **Comparison with other multi-trackers.** This table shows the comparison of our muti-tracker with respect to others (tierpsy-tracker [33], WF-NTP.v3 [38])

Comparative table				
Name	Illumination technique	Features	Method	Situations solved
Tierpsy Tracker [33]	Standard backlight	Manual parameter setting	Skeletons, outlines and segmentations	Individual tracking
WF-NTP.v3 [38]	Flat-field illumination	Manual parameter setting	Skeletons and centroids	Individual tracking
Ours	Active backlight system [64, 67]	Fully automatic	Improved skeleton and segmentations	Individual tracking, overlaps, body contacts, rolled worms and occlusions.

2.2.3 Materials and Methods

Nematode Strains and Culture

The strains (N2) and CB1370 *daf-2* (e1370) used in the study were provided by the University of Minnesota Caenorhabditis Genetics Center. The *C. elegans* were kept at 20ºC and cultured on 55mm diameter NGM plates with 1 µg/mL of fungizone to prevent fungal contamination [88]. Escherichia coli (*E. coli*) strain OP50 was used as standard prey for of *C. elegans* in the laboratory. FUdR (0.2 mM) was used to prevent replication. The cultured plates were 10, 15, 30, 60 and 100 adult-young worms synchronized from worm eggs incubated at 20ºC. This variety in the number of worms per plate allowed to obtain aggregations of two *C. elegans* or more.

Proposed tracking method

At present, automatic or semi-automatic trackers discard the data of tracks where there are overlap or body contacts, due to the difficulty of solving the identity of each individual in these situations. Our method does solve these situations using a fully automated pipeline based on an intelligent active backlight, a modified skeletonization method and new criteria to solve the final optimization problem.

Steps of the proposed tracking method are described in Figure 2.13a, b, c, d, e, f, and more detailed in subsections 2.3, 2.4, 2.5, 2.6, and 2.7 respectively. This method starts with the acquisition of sequences of *C. elegans* images, Figure 2.13a. These image sequences are then processed (Figure 2.13b) to obtain mathematical models of *C. elegans* Figure 2.13c, and possible solutions to these in an aggregation. Finally, the result (Figure 2.13f) is the optimal minimization cost between models and possible solutions.

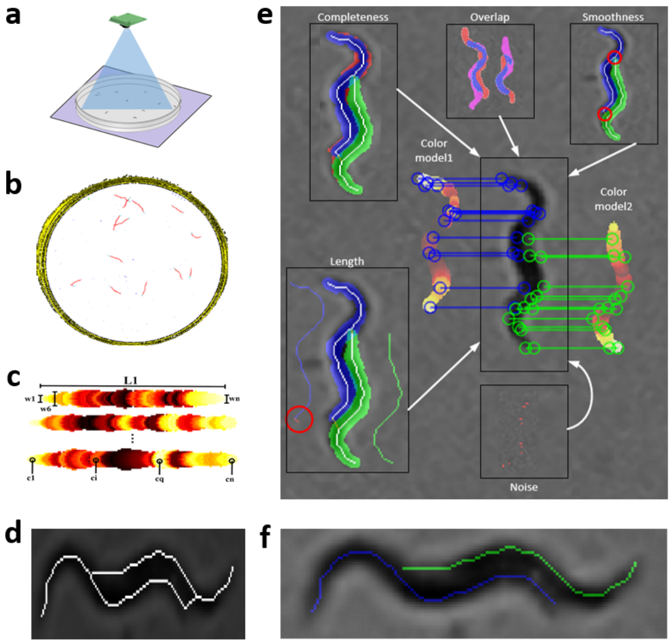


Figure 2.13: **General scheme of image processing.** The image shows the different stages that the tracker goes through to obtain the results of the skeletons and to follow all the worms within the plate. **(a)** Image acquisition [67]. **(b)** Pre-processing image. **(c)** Worm models. **(d)** Improved skeleton using proposed method [41]. **(e)** Optimization method. **(f)** Optimization results (predicted poses).

Image acquisition

The image acquisition process was performed using the capture system [67]. To use this system, first, a laboratory operator removed the plates from the incubator, then the plates were analyzed to find condensation on the covers, if so, they were removed, otherwise, the image sequence was captured. The system [67] is automatic and captures an image every second. *Escherichia coli* (*E. coli*) strain OP50 was placed in the center of the plate to capture the worms inside the Petri dish and not scaling the edges or near them.

The intelligent active backlighting method was used as the illumination technique [64]. This method is more robust than standard backlighting methods, it allows to obtain constant intensity values for the bottom of the Petri dish and the worms (greater than 48 and less than 35 respectively). This facilitated automatic segmentation with fixed thresholds on all images.

Table 2.4: **Components of the data acquisition system.**

System Components		
Nº	Name	Description
1	Raspberry Pi V3 b+	Procesador : 64-bit ARM Cortex-A53, 1.4GHz RAM Size: 1GB LPDDR2 SDRAM
2	Raspberry Pi Camera V1.3	Sensor: OmniVision OV5647 Pixel resolution: 2592 x 1944 Pixel size: 1.4 x 1.4 µm Field of view: 53.50º x 41.41º Optical size: 1/4 ″ Focal length: 2.9
3	Raspberry Pi display	Screen display size: 7″ Resolution: 800 x 480 and 60 fps Color: 24-bit RGB colour

The image sequence was acquired at a resolution of 1944 x 1944 pixels and a frequency of 1 Hz (1 image per second) using the system [67] (Table 2.4).

This image acquisition system is open hardware and its assembly procedure, parts and description are described in detail in another work [67].Worm tracking, using these image acquisition conditions, is a very challenging problem. An image resolution of 1944x1944 pixels is the lowest able to detect worms, when a complete Petri dish (55 mm. of diameter) is monitored with a fixed camera. In addition, this problem was solved by using a low frame rate of 1Hz. The dataset collected was composed by sequences of 30 images where contact events between worms occurred, to perform all the experiments.

Image processing

Image processing began with the segmentation of the region of interest and e *C. elegans* tracks in the image sequence (white circle and red track in Figure 1b). To obtain the region of interest, first, a segmentation was carried out on all the images of the sequence using a threshold with a fixed intensity value (35 in the gray scale). Then an AND operation was executed between all the segmented images, the result obtained went through a "Fillhole" operation to fill small holes. The region of interest was selected as the largest connected component of the resulting image. It is important to mention that the use of a fixed threshold for all images is due to the intelligent active backlighting method as a lighting technique [64], this system allows to conserve background intensity values constantly.

In parallel to the previous step, the *C. elegans* tracks were segmented with a process using different threshold levels Figure 2.13b. Threshold levels were below 35 on the gray scale. The resulting segmentations went through two filters in order to eliminate those tracks that did not correspond to worms. In the first stage, those tracks with an area smaller than the minimum area of a worm were filtered. The second filter analyzed the skeleton of each image, if the skeleton did not correspond to a minimum expected length it was classified as noise. The results of number of skeletons found in each image were stored in a 30xN matrix, where N is total number of tracks and 30 is the number of images in the sequence.

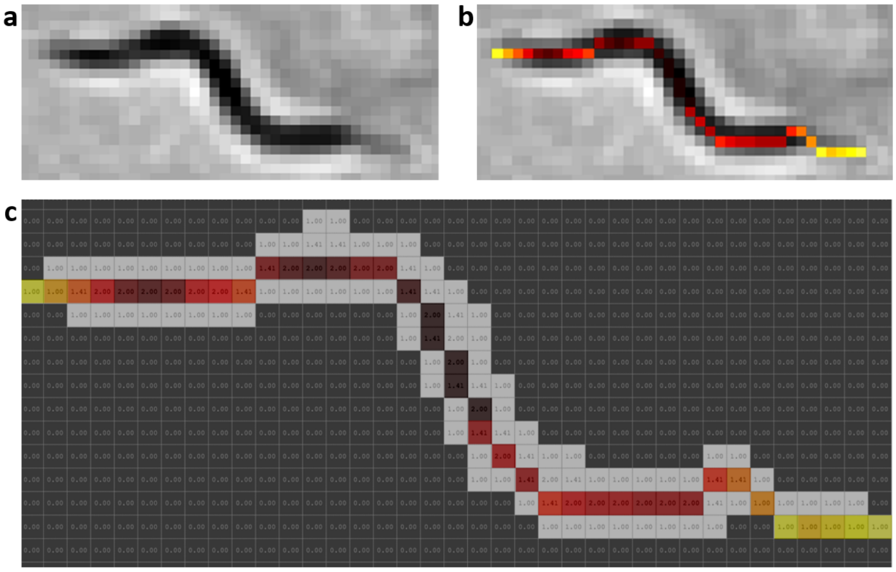
Due to low resolution of the worms, a scale factor of 3 was applied to increase resolution. This process was applied by obtaining the model to the end of tracking. The model of each worm was obtained by analyzing 30xN matrix, finding image "k" of 30 images, where worms were further apart from each other, and their ends, head-tail, were also separate. Tracking of each worm started from image k+1 to image 30 and from image k-1 to the first. The skeletonization method proposed in previous work [41] was used in each image. This method used distance transformation [81] to obtain possible worm skeletons, and through of an optimization method using different criteria found the best skeleton prediction. Once the tracking process had finished, the results were reconverted to the original scale to be saved.

Worm model

At present, there are different skeletonization methods [81, 85]. Matlab's bwmorph function was used as a classic skeletonization method to obtain worm skeleton model (color pixels in Figure 2.14b). The proposed worm model consists of width and color values along the skeleton of each individual, Figure 2.14c. The width values are obtained using classical skeleton in resulting image after using distance transformation function on segmentation of the image "k". Grayscale image show in Figure 2.14a. While color values of pixels are obtained using the classical skeleton in gray image, Figure 2.14b. The length value is total number of pixels in the skeleton. The length model is averaged while the *C. elegans* are separate and tracking progresses.

Extraction of possible solutions

The skeletonization method proposed in the previous work [41], unlike classical methods, enables the separation of aggregated worms (Figure 2.15a), creating new paths in the skeleton, and some possible solutions for each worm (Figure 2.15b, c, d, e). Maximum and minimum values of the width vector are used in the distance to transform images of each segmentation to find this new skeleton as mentioned in [41]. The possible solution skeletons are obtained from a recursive function, which runs through ends and branch points of the new skeleton that overlap the previous segmentation of the worm's body (red circle in Figure 2.15b, c, d, e).



0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.41	1.41	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.41	2.00	2.00	2.00	2.00	2.00	1.41	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	1.41	2.00	2.00	2.00	2.00	2.00	2.00	1.41	1.00	1.00	1.00	1.00	1.00	1.00	1.41	1.41	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	2.00	1.41	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.41	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.41	1.41	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.41	1.41	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	2.00	1.41	1.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.41	2.00	1.41	1.00	1.00	1.00	1.00	1.00	1.41	1.41	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.41	2.00	2.00	2.00	2.00	2.00	2.00	1.41	1.00	1.00	1.00	1.00	1.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 2.14: **Prediction model.** Skeleton gray values were changed with a HOT color map for better visualization. (a) Grayscale image. (b) Color values obtained from grayscale image. (c) Values of widths marked with the colors of the model; the length is the total pixels in the skeleton.

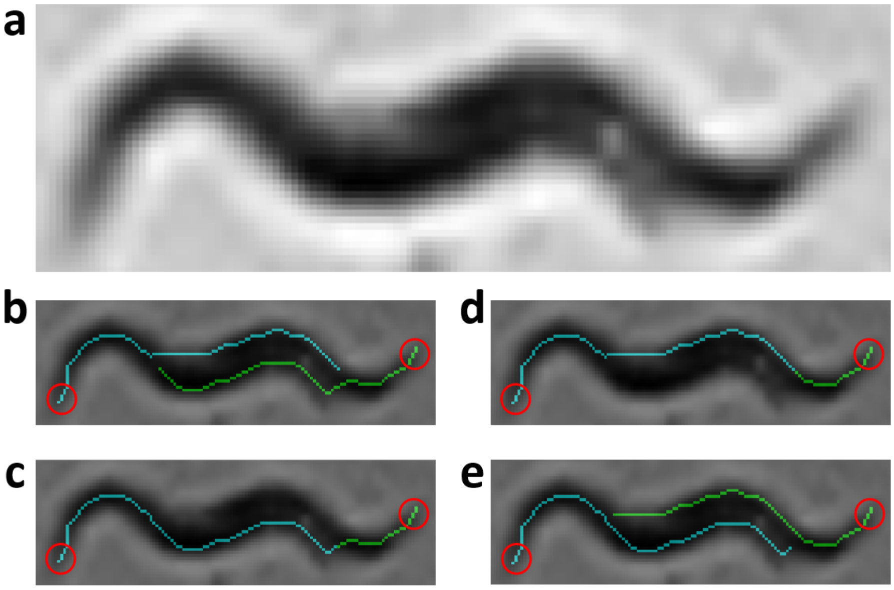


Figure 2.15: **Possible solutions.** Circles in red mark the starting point to find the possible skeleton. Cyan and green lines are the possible solutions for each worm. **(a)** Grayscale image. **(b, c, d, e)** Possible solutions.

Optimization method

The prediction "S" of the following postures will be the possible solution with the minimum value within all the possible "P" combinations of skeletons in one segmentation (2.2). The value of each possible combination "C_P" of skeletons is obtained from the sum of criteria "m", for the number of worms "n" in an aggregation (2.3). The criterion "C_p" is evaluated for each possible worm "i" and for each criterion "j". The criteria analyzed were the length of skeleton, overlap with the previous body, completeness, smoothness of the skeleton, noise in segmentation and the colors of each worm.

$$S = \underset{p}{\arg\min}(C_p) \tag{2.2}$$

$$C_p = \sum_{j=1}^{m} \left(\sum_{i=1}^{n} C_j^i \right) \tag{2.3}$$

The length and color criteria prevent the prediction differing from the model in length and color. The smoothness criterion prevents sudden changes in the direction of the skeletons. The completeness criterion prevents the current segmentation from being incomplete. The overlap criterion with the previous body prevents the identity change during aggregation. And the noise criterion prevents the skeleton prediction falling on the plate noise.

The reconstruction of the body of each worm was used for the evaluation of the different criteria. This was performed by using the skeleton pixels in each possible prediction with the width and color values obtained in the model (prediction start), respectively for each individual.

Length criterion

The length criterion "C_L" is obtained from the sum of the multiplication of average squared width $[\overline{W_i^2}]$ with the difference in length (Δ_i), (2.4). This difference is obtained from subtraction between the model length of each worm (L_i) and length of the skeleton obtained (WL_i). The average squared of the width was used so that the resulting length criterion is as significant as the rest of the criteria where the error in pixels was evaluated. Figure 2.16a shows the aggregation of two worms in a gray scale, and Figure 2.16b, c shows possible posture predictions where length errors occur.

$$C_L = \sum_{i=1}^{n} \left(\Delta_i \overline{W_i^2} \right) \tag{2.4}$$

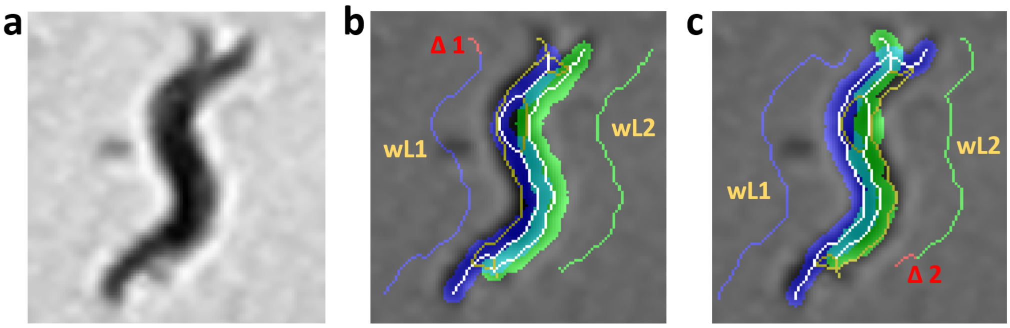


Figure 2.16: **Length criterion evaluation.** The pixels in red are the length error. The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). **(a)** Grayscale image. **(b)** Length criterion evaluation with length error in worm1 (blue). **(c)** Length criterion evaluation with length error in worm2 (green).

Overlap criterion

The overlap criterion “C_O” is obtained from the sum of the absolute difference of the reconstruction of the worm’s body in the previous state, B_{Ppx}, (white dashed line in Figure 2.17b, c, e, f) and the current state, B_{Cpx}, (green and blue segmentation in Figure 2.17b, c, e, f) for each possible worm "i" in the aggregation (2.6) (2.6). This is done for all “m” pixels in each reconstruction. Figure 2.17a shows in gray scale the aggregation of two worms, Figure 2.17d, g shows the next postures predictions (blue and green segmentation), parallel to these in dashed lines showing the previous state (B_{Ppx}) and the rest show the current state (B_{Cpx}). In Figure 2.17d it can be seen that the overlap criterion is low because it belongs to the same worms, while in Figure 2.17g it is higher due to the identity change.

$$C_O^i = \sum_{px=1}^{m} \begin{cases} 1 & if \ \left|B_{Ppx} - B_{Cpx}\right| == 1 \\ 0 \end{cases} \tag{2.5}$$

$$C_O = \sum_{i=1}^{n} C_O^i \tag{2.6}$$

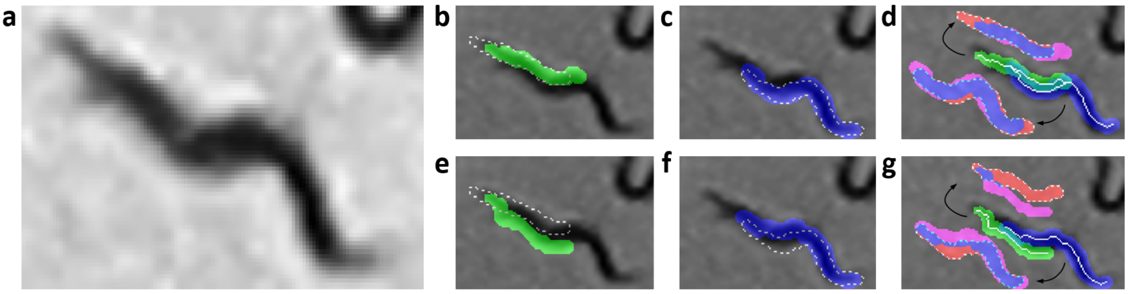


Figure 2.17: **Overlap criterion evaluation.** The pixels in red and magenta are the error of overlap with the previous state. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). The dashed line segmentation is the previous state segmentation. **(a)** Grayscale image. **(b, e)** Previous state (Bps_{px}) in white dashed line and current state (Bcs_{px}) in green. **(c, f)** Previous state (Bps_{px} in white dashed line and current state (Bcs_{px}) in blue. **(d)** Evaluation with low overlap criterion. **(g)** Evaluation with high overlap criterion.

Completeness criterion

The completeness criterion “C_{Cp}” is obtained from the sum of the absolute difference of the current segmentation of the image (A_S) and the reconstruction of each possible worm body "i" in the current state (B_{Cpx}) (2.7) (2.8). This is done for all “m” pixels in each reconstruction or current segmentation. Figure 2.18a shows the aggregation of two worms in gray scale, Figure 2.18b, c, d shows completeness values for each image. Figure 2.18b is the correct prediction, while Figure 2.18c shows incorrect prediction, due to the identity change and final extremes. Figure 2.18d shows low completeness criterion and an incorrect prediction too, due to the change of identity in both worms.

$$C_{Cp}^i = \sum_{px=1}^{m} \begin{cases} 1 & if \ \left|A_{Spx} - B_{Cpx}\right| == 1 \\ 0 \end{cases} \tag{2.7}$$

$$C_{Cp} = \sum_{i=1}^{n} C_{Cp}^i \tag{2.8}$$

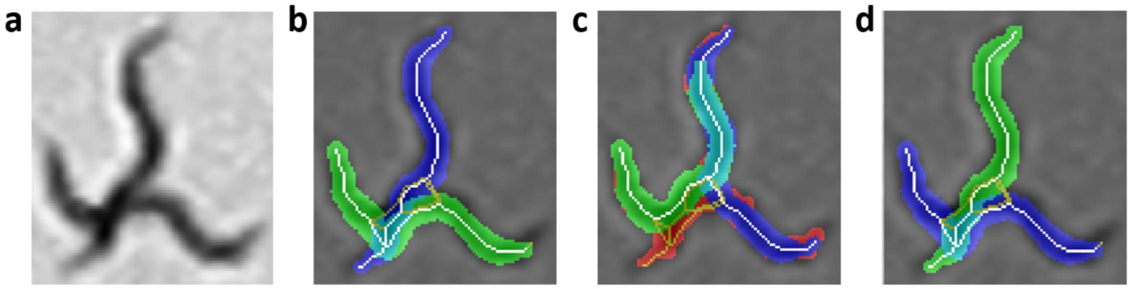


Figure 2.18: **Completeness criterion evaluation.** The pixels in red are the completeness error. The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). **(a)** Grayscale image. **(b)** Correct prediction with low completeness criterion. **(c)** Incorrect prediction with high completeness criterion. **(d)** Identities changed and with the same completeness criterion as image b.

Smoothness criterion

The smoothness criterion “C_S” is obtained from the average of the absolute values of the angles obtained for the "nk" pixels of the skeleton and for each worm "i" in the segmentation (2.10). For each pixel of the skeleton there is an angle (θ_{px}), which is obtained by an average of the sum of "nA" angles before and "nA" angles after divided by the total pixels found "c" (2.9). The value of "c" will be 2*nA if there are "nA" pixels before and after the pixel to be evaluated (2.9). Figure 2.19a shows the aggregation of two worms in gray scale, Figure 2.19b shows a possible prediction of skeletons with low softness criterion, while Figure 2.19c, d show possible predictions of skeletons with high softness criterion.

$$\theta_{px} = \frac{\sum_{a=x-nA}^{x+nA} \theta_a}{c} \tag{2.9}$$

$$C_S = \sum_{i=1}^{n} \left(\sum_{px=1}^{nk} \frac{|\theta_{px}|}{nk} \right) \tag{2.10}$$

Noise criterion

The noise criterion “C_N” is obtained from the intersection of the noise segmentation (N_S) and the reconstruction of each body of a possible worm "i" in the current state (B_{Cpx}) (2.11) (2.12). This is done for all “m” pixels in each reconstruction or noise segmentation. Figure 2.20a shows the aggregation of one worm with noise in a gray scale, while Figure 2.20b shows the correct prediction (segmentation in blue) with a low noise criterion (segmentation in magenta). Figure 2.20c shows an incorrect prediction (blue-magenta segmentation) with a high noise index (magenta segmentation).

$$C_N^i = \sum_{px=1}^{m} \begin{cases} 1 & if \ \left|N_{Spx} \& B_{Cpx}\right| == 1 \\ 0 & \end{cases} \tag{2.11}$$

$$C_N = \sum_{i=1}^{n} C_N^i \tag{2.12}$$

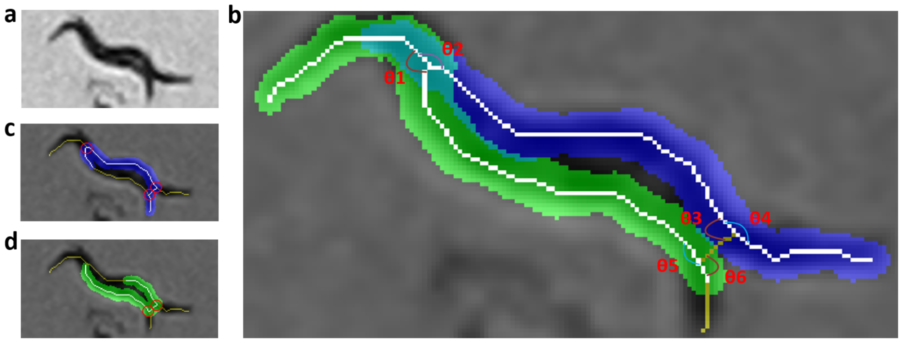


Figure 2.19: **Smoothness criterion evaluation.** The yellow and white pixels are the resulting skeleton using the improved skeletonizing method. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). **(a)** Grayscale image. **(b)** Evaluation of smoothness, angles in red are those that have more weight and increase the index of smoothness. **(c)** Incorrect prediction of worm1 with high softness criterion. **(d)** Incorrect prediction of worm2 with high softness criterion.

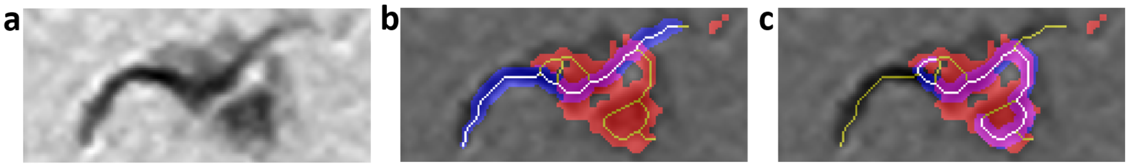


Figure 2.20: **Noise criterion evaluation.** The pixels in magenta (intersection of blue and red) are the noise error. The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the worm's body (segmentation in blue). **(a)** Grayscale image. **(b)** Correct prediction with low noise criterion. **(c)** Incorrect prediction with high noise criterion.

Color criterion

The color criterion "C_{Cl}" is obtained by adding all the pixels with an absolute difference greater than the threshold value (U = 1) of the color model (C_M) with respect to the reconstruction of the current body (C_A) (2.13), for each worm in the aggregation (2.14). This means that the error only increases the value by 1 if the absolute difference between a pixel of the reconstruction of the body of the possible worm "i" using the color model (C_{Mpx}) and the reconstruction of the same worm "i" in the current segmentation (C_{Apx}) is greater than the threshold value. This is done for all "m" pixels in the reconstruction. For both worm body reconstructions (model and current prediction) the same width and length values are used, in order to compare pixels 1 to 1. Figure 2.21a shows the aggregation of 2 worms in gray scale, while the Figure 2.21b, c shows the comparison of each prediction with its respective model.

$$C^i_{Cl} = \sum_{px=1}^{m} \begin{cases} 1 & if \ \left|C_{Mpx} - C_{Apx}\right| > U \\ 0 \end{cases} \tag{2.13}$$

$$C_{Cl} = \sum_{i=1}^{n} C^i_{Cl} \tag{2.14}$$

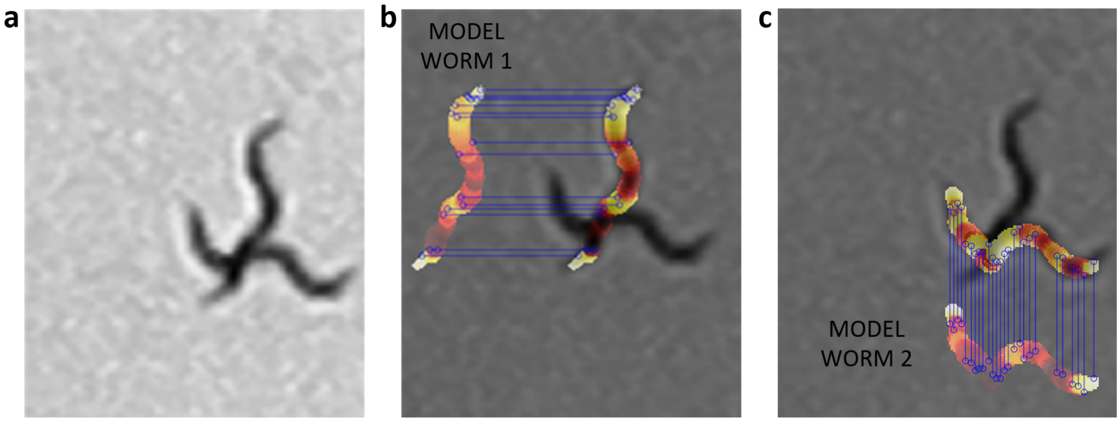


Figure 2.21: **Color criterion evaluation.** For each possible skeleton, its color values are obtained using current grayscale image, with those color values and the width values of the model each worm is reconstructed and compared with the reconstruction of the model (model worm). The gray values were changed for a HOT color map in order to better visualize them. The blue lines indicate those pixels different from the model. **(a)** Grayscale image. **(b)** Comparison of worm1 model with current prediction of worm1. **(c)** Comparison of worm2 model with current prediction of worm2.

Evaluation method

Manually labeled skeletons were used as a reference to compare all the results. These skeletons were obtained using an application designed to select each pixel belonging to the skeleton of each worm one by one in the image sequence. This operation was performed for all 3,779 postures of the 70 plates used. The shape of these nematodes was recovered using a disk-shaped dilation operation of radius equal to half the width (approx. 2 pixels) on the skeletons obtained.

The Jaccard coefficient, or intersection over the union (IoU), was used to measure the degree of precision in locating worms (2.15). As its name indicates, it is obtained by dividing the total area of the intersection by the union of the elements [39]. For the evaluation, the area of the reconstructed bodies of the manually labeled skeletons was used, skeletons using the skeletonization method proposed in [41] and the classical skeletonization method using the Matlab bwmorph command.

$$IoU = \frac{\sum P_{w1} \bigcap P_{w2}}{\sum P_{w1} \bigcup P_{w2}} \tag{2.15}$$

The IoU index was expected to be higher because a predicted pose (Figure 2.22b) is compared to an annotated ground-true pose (Figure 2.22a) which must overlap (Figure 2.22c, d, e). The results for the example below are IoU = 0.9784, 0.5667, and 0.2649, respectively.

Matlab 2018b Machine Learning Toolbox was used to obtain the comparison statistics between prediction models (Supplementary Figure A.17a, b, Figure A.18a, b, Figure A.19a, b, Figure A.20a, b) and the two skeletonization methods (Supplementary Figure A.21a, b, Figure A.22a, b). The Kolmogorov Smirnov normality test was used for large samples (n>50) and the Wilcoxon Signed Ranks test.

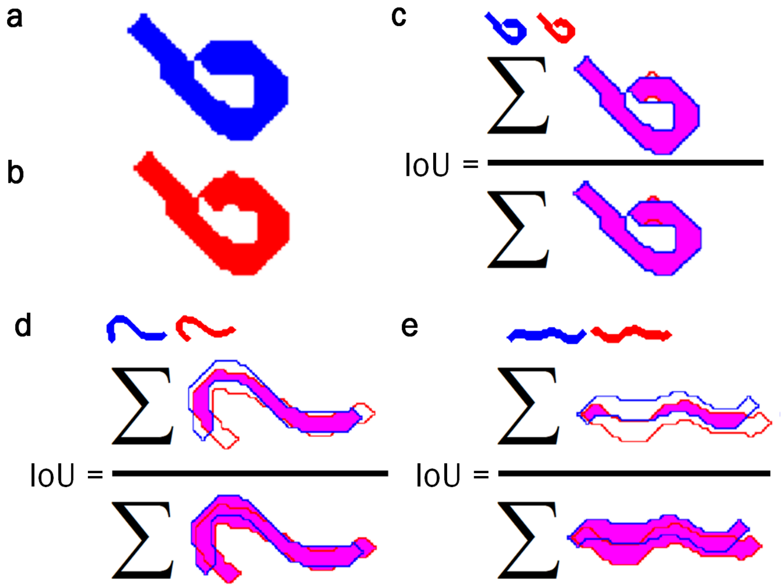


Figure 2.22: **IoU index.** This index evaluates how close the response of the automatic method comes to the reference, and compares both results to determine the improvement of one method over the other. The higher the IoU value, the closer the response comes to the reference. The evaluation is performed by reconstructing the skeletons obtained with a radio 2 disk. **(a)** Reconstructed body of the manually labelled skeleton. **(b)** Reconstructed body of the skeleton obtained using the new skeletonization method [41] or the classical method. **(c, d, e)** Evaluation of the reconstructed skeletons.

2.2.4 Results

Experiments were performed with 70 plates. Of these, 54 corresponded to plates with 10 and 15 worms, one plate with 30 worms, four plates with 60 worms, and 11 plates with 100 worms, totaling 65400 worm poses. All these data were analyzed to obtain with contact between worms and noise. As demonstrated in [89, 66], a higher number of worms per plate will increase the likelihood of contact between them. Nematodes studied were young-adult wild-type (N2) and CB1370, *daf-2* (e1370), as mentioned above.

53 tracks with 3,240 poses were used to evaluate aggregation between worms, and 17 tracks with 509 poses for aggregation between worms and noise. The IoU index was used to evaluate the percentage of success in tracking the worms and also to compare both skeletonization methods. The area of worm bodies reconstructed from skeletons obtained manually and skeletons obtained with the two skeletonization methods (new and classical) was used to evaluate the IoU index.

Different prediction models were implemented in order to find the most significant criteria. The name of each model has been coded using letters from the criteria names. "O" for overlap, "L" for length, "Cp" for completeness, "N" for noise, "S" for smoothness, and "Cl" for color. The model with the best results was model7 (OCpCl) with a 99.42% percentage accuracy in aggregated worm tracks and an IoU value of 0.70 in average. Figure 2.23a, b, c, d, e, shows an example using the model7, in this image you can see the evaluation of the 3 criteria of this model and optimization result. Some examples of aggregation cases are presented at the end of the supplementary material using mode7 (Supplementary Figure A.23, A.24, A.25, A.26, A.27).

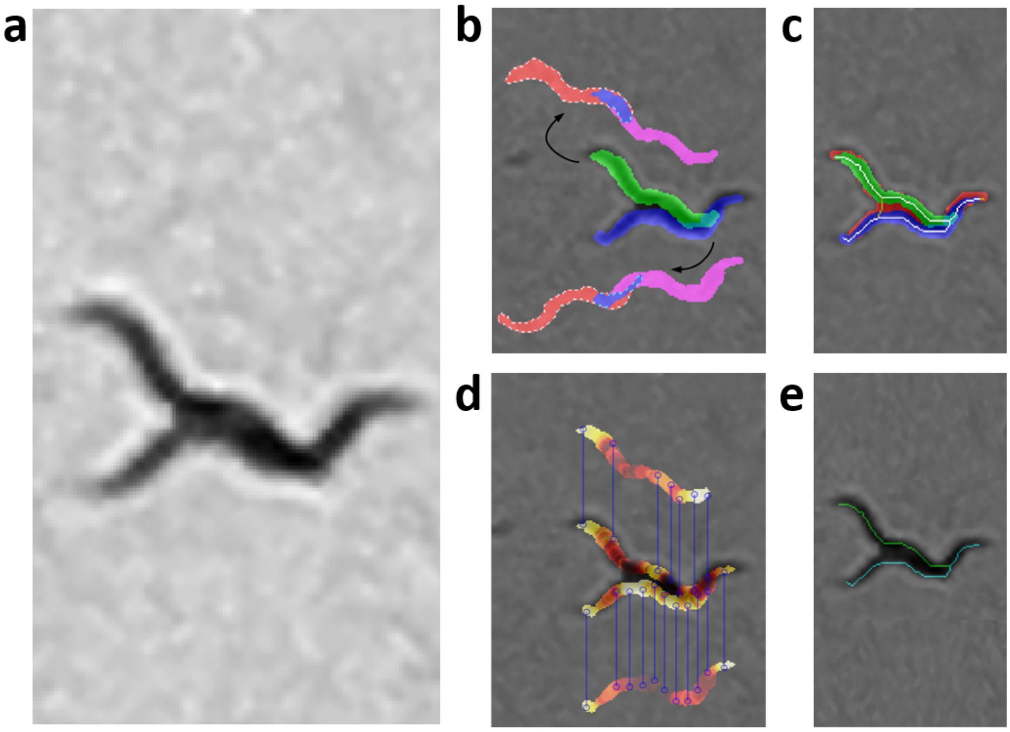


Figure 2.23: **Model7 evaluation.** The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). **(a)** Grayscale image. **(b)** Overlap criterion evaluation. **(c)** Completeness criterion evaluation. **(d)** Color criterion evaluation. **(e)** Optimization result.

To measure the percentage accuracy in tracking the worms, non-zero IoU values were used, from the beginning to the end of the tracks. The accuracy of the results of the different prediction models using the 2 skeletonization methods are shown in Table 2.5. In addition, the average IoU value was obtained for all the prediction models (see Table 2.6), from the beginning of the aggregation to the end. 790 poses were used in aggregations between worms and 509 poses for worms aggregated with noise, 1299 poses in total.

Table 2.5: **Comparative table of percentage accuracy of postures for models and methods.** The table shows the percentage accuracy of poses (skeletons) for each model and method used during the tracking *C. elegans*. 3240 poses were used to evaluate tracks where there is aggregation of two or more worms, and 509 poses to evaluate the aggregation between worms and noise on the plate.

Worms aggregation			
Nº	Model	New	Classical
1	O	98.49	97.80
2	OL	98.15	97.99
3	OCp	99.13	98.09
4	ON	98.09	97.88
5	OS	98.07	97.72
6	OCl	98.57	98.20
7	OCpCl	**99.42**	**98.73**

Table 2.6: **Summary of model and method comparison.** This table shows the results for each prediction model and two skeletonization methods, New [41] and classical (using the Matlab bwmorph function). The results column indicates the percentage value of the improvement using the new skeleton with respect to the classic skeletonization.

			Average IoU		Standard Deviation		Results %
N°	Model	Total pose	New	Classical	New	Classical	Improvement
1	O	1299	0.64	0.63	0.23	0.22	0.39
2	OL	1299	0.63	0.63	0.23	0.22	0.10
3	OCp	1299	0.69	**0.67**	0.19	0.19	2.61
4	ON	1299	0.63	0.63	0.24	0.22	0.50
5	OS	1299	0.63	0.63	0.23	0.22	0.04
6	OCl	1299	0.65	0.65	0.23	0.20	-0.15
7	OCpCl	1299	**0.70**	**0.67**	0.19	0.18	**2.66**

Comparison with other trackers

At present, automatic or semi-automatic trackers discard the data of tracks where there are overlap or body contacts, due to the difficulty of solving the identity of each individual in these situations. Our method does solve these situations so a direct comparison cannot be made with the results of other trackers.

To compare our method graphically with other trackers (tierpsy-tracker [33], WF-NTP.v3 [38]), labeled data was first shaded in different grays and overlapped predicted data in colors (Figure 2.24a, b, c). Errors for each comparison are shown in grayscale. The comparison was made with bodies reconstructed from skeletons (skeleton dilation with disk equal to 2), except for WF-NTP.v3 [38] whose results are skeleton centroid points. The tierpsy-tracker [33] multi-tracker, Figure 2.24a, did not resolve path 2 where aggregation occurs, while the other tracks (1, 3, 4) were partially resolved, due to the occlusions that the worms make on themselves. The multi-tacker WF-NTP.v3 [38], Figure 2.24b, did not solve the identity problem in track 2. And like the previous multi-tracker it presented problems in the tracks with occlusions (1, 3, 4). Model 7, on the other hand, had almost zero errors, tracked all worms, resolved the identity in the aggregation of track 2 and the occlusions in the remaining tracks, Figure 2.24c.

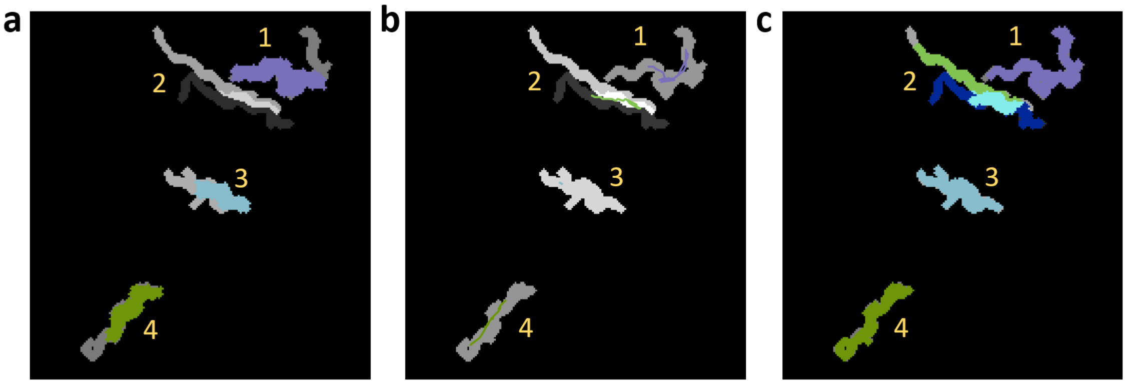


Figure 2.24: **Comparison of trackers.** Comparison of reconstruction of *C. elegans* bodies between labeled data (shaded in grays) and predictions obtained with different trackers (shaded in colors). **(a)** Results obtained using tierpsy-tracker [33]. **(b)** Labeled data (grays) compared with colored lines which connect centroids obtained using WF-NTP.v3 [38]. **(c)** Results obtained using model7.

2.2.5 Discussion

Caenorhabditis elegans are aggregated in different ways, such as aggregation of end parts (head or tail), partial aggregation of bodies, aggregation of parallel bodies, among others. The experiments were conducted with the above mentioned methods with a single criterion and found that the overlapping criterion with the previous prediction is the most significant. This criterion allows part of the previous state to be preserved, helping to solve the next state. A result of 98.49% was obtained using this criterion individually. However, when aggregation gives rise to an overlap in most worms, it is difficult to identify them, even for human observers. To solve this problem, different tests were designed, combining the criteria mentioned in the optimization methods section.

The completeness in cases of aggregation, where there are overlaps between aggregated bodies, partial and total identity changes could be obtained, as observed in the optimization method Figure 2.18c, d. When using the overlap criterion with completeness criterion, a percentage accuracy of 99.13% was achieved. As shown in supplementary material (Supplementary Figure A.17a, b, Figure A.18a, b) this criterion is statistically significant, helping to solve some cases where the overlap criterion presented problems, Figure 2.17b.

The color criterion also helped to improve worm prediction, although not so significant (Supplementary Figure A.19a, b, Figure A.20a, b). The end portion of *C. elegans* (tail) has a greater number of light pixels (higher gray levels) than the head, where there are fewer. These features are important because before, during and after an aggregation one of the final parts remains visible, and with this color singularity a more accurate prediction can be obtained, helping to solve the identity problem after an aggregation between worms or noise. The use of these three criteria (overlap, completeness, and color) allowed us to obtain a percentage accuracy of 99.42%.

The length, smoothness, and noise criteria were discarded because when they were used individually or together with the overlap criterion, their percentage accuracy decreased. On the one hand, the problem with the smoothness criterion was due to the low resolution of the worms in our images, which provided a poor estimate of this criterion. On the other hand, the problem with the length criterion was due to errors in the length model and the increase or decrease in the length of worms owing to overlaps. The problem with the noise criterion was due that when the worm is visualized against background noise, many possible solutions were generated.

Finally, it is worth mentioning that the best combination of criteria depends on image quality. In this work, we demonstrated that the combination of the three criteria mentioned above (overlap, completeness, and color) was the best option for automatic tracking of interaction behaviors among *C. elegans* (contacts or overlapping) with our low-resolution dataset.

2.2.6 Conclusions

This paper presents a method for tracking multiple *C. elegans* in standard Petri dishes where some worms can come into contact or overlap. This method was evaluated in a difficult scenario using a low-image resolution and a low frame rate. Using an optimizer with the appropriate criteria (overlap, completeness, and color) was shown can solve many worm overlap and contact situations. The accuracy obtained under these conditions was 99.42 % and 98.73 % using the modified skeleton algorithm and the classical skeleton algorithm respectively.

In addition, the proposed method employs an improved active backlight system and an improved skeletonization algorithm. The active backlight system provides fixed gray levels in all captured images, which allows automatic segmentation using a fixed threshold. The improved skeletonization algorithm uses width information from each worm model to extract skeletons, enabling the tracking of worms moving in parallel (side by side).

Our proposal, unlike other trackers that discard worm overlaps and contacts, solves many of these situations, increasing the number of worms tracked in a Petri dish and therefore paving the way to automate new assays where interaction between worms occurs.

Author Contributions: conceptualization, P.E. Layana Castro and A.J. Sanchez-Salmeron; methodology, P.E. Layana Castro and A.J. Sanchez-Salmeron; software, P.E. Layana Castro; validation, P.E. Layana Castro, J.C. Puchalt, A.J. Sanchez-Salmeron and A. García Garví; formal analysis, P.E. Layana Castro and A.J. Sanchez-Salmeron; investigation, P.E. Layana Castro and A.J. Sanchez-Salmeron; resources, A.J. Sanchez-Salmeron; data curation, P.E. Layana Castro, J.C. Puchalt and A. García Garví; writing–original draft preparation, P.E. Layana Castro and A.J. Sanchez-Salmeron; writing–review and editing, P.E. Layana Castro, J.C. Puchalt, A. García Garví and A.J. Sanchez-Salmeron; visualization, P.E. Layana Castro and A.J. Sanchez-Salmeron; supervision, A.J. Sanchez-Salmeron; project administration, A.J. Sanchez-Salmeron; funding acquisition, A.J. Sanchez-Salmeron. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Plan Nacional de I+D with Project RTI2018-094312-B-I00, FPI Predoctoral contract PRE2019-088214 and by European FEDER funds.

Data Availability Statement: The programme was developed in Matlab2018b with Windows 10 and works correctly in later versions with Image Processing, Communications and Bioinformatics toolbox. Its source code is in GitHub. It is open-source MIT (Massachusetts Institute of Technology) and can be downloaded from the repository at https://github.com/playanaC/WT_ISA. The dataset with all aggregation experiments can be downloaded from https://active-vision.ai2.upv.es/wp-content/uploads/2021/02/dataset_skeletons.zip.

Acknowledgments: ADM Nutrition, Biopolis S.L. and Archer Daniels Midland supplied the *C. elegans* plates. Some strains were provided by the CGC, which is funded by NIH Office of Research Infrastructure Programs (P40 OD010440). Mrs. Maria-Gabriela Salazar-Secada developed the skeleton annotation application. Mr. Jordi Tortosa-Grau annotated worm skeletons.

Conflicts of Interest: No conflict of interest exists.

2.3 Paper3

Caenorhabditis elegans detection using YOLOv5 and Faster R-CNN networks

Ernesto Jesús Rico Guardiola[1], **Pablo E. Layana Castro**[1], **Antonio García Garví** [1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Rico-Guardiola, E.J., Layana-Castro, P.E., García-Garví, A., Sánchez-Salmerón, AJ. (2022). *Caenorhabditis Elegans* Detection Using YOLOv5 and Faster R-CNN Networks. In: Optimization, Learning Algorithms and Applications. OL2A 2022. Communications in Computer and Information Science, vol 1754. Springer, Cham. https://doi.org/10.1007/978-3-031-23236-7_53.

2.3.1 Abstract

The detection of *Caenorhabditis elegans* (*C. elegans*) is a complex problem due to the variety of poses they can adopt, the aggregations and the problems of dirt accumulation on the plates. Artificial neural networks have achieved great results in detection tasks in recent years. In this article, two detection network architectures have been compared: YOLOv5s and Faster R-CNN. Accuracy and computational cost have been used as comparison criteria. The results show that both models perform similar in terms of accuracy but the smaller version of the YOLOv5 network (YOLOv5s) is able to obtain better results in computational cost.

keyword: *C. elegans* - detection network - YOLOv5; Faster R-CNN.

2.3.2 Introduction

The nematode *Caenorhabditis elegans* (*C. elegans*) has been used as an important animal model for many years [5]. In advanced ages, just like humans, they present some pathologies, physical deterioration, mobility loss, etc. It has a short life (approximately 3 weeks) and a length of 1 mm, which makes it easy to handle and grow. They can be stored in large quantities and small spaces (Petri dishes) [58]. These characteristics allow large-scale assays to be performed economically and make this nematode an ideal model for research into neurodegenerative diseases and the development of new drugs.

Despite these advantages, handling and observation of these nematodes are costly and require qualified technicians. For this reason, computer vision techniques allow these processes to be automated, increasing productivity and precision in the development of assays.

The detection of *C. elegans* is the first task in automatic image processing applications such as monitoring, tracking [38, 33, 41, 42], lifespan [21, 65] and healthspan [23, 44, 14]. To date, many methods perform segmentation pre-processing and search for worm characteristics in order to filter worms (object of interest) from non-worms (noise), which takes an additional computational cost.

In recent years, image processing methods based on artificial neural networks have gained much popularity in these *C. elegans* applications, as they have great advantages compared to traditional computer vision methods based on feature extraction (using segmentation, edge detection, textures). Feature extraction is a costly process as it requires experience and trial and error to select the optimal ones. In addition, it is difficult to adjust the parameters of the image processing algorithms to work on different images.

Detection methods based on artificial neural networks have achieved great results, outperforming classical methods. The main detection methods based on deep learning can be classified into two groups: one-stage methods (YOLO, SSD, RetinaNet) and two-stage methods (Faster R-CNN, RFCN, Mask R-CNN).

These are some of the latest studies that show the advantages of artificial neural networks in the detection of *C. elegans*:

- [4] Implement Faster R-CNN object detection models for detection and location of *C. elegans* in different conditions. This study demonstrates some of the advantages of Faster R-CNN algorithm to identify, count and track behaviors in comparison to traditional methods in terms of speed and flexibility. These traditional image processing methods, such as background subtraction or using morphological features, are more sensitive to changes in the environmental conditions due to complex experimental setups, resulting in low accuracy to detect objects of interest.
- [20] Present a low-cost system including a DIY microscope to capture low-resolution images. They use the framework Mask R-CNN to detect, locate and classify *C. elegans* with high accuracy, rather than traditional methods such as human counting and analyzing high-resolution microscope images, which result in longer times and lower efficiency. They also generate a great number of masks by a preprocessing method that includes visual inspection, since the Mask R-CNN algorithm needs them as an input.

In summary, the main goals of this study are:

- Analyze the use of detection methods based on artificial neural networks for the detection of *C. elegans* in low resolution images.
- Compare two of the most widely used object detection architectures (YOLO [71] and Faster R-CNN [72]) in terms of accuracy and computational cost.

2.3.3 Materials and method

C. elegans Strains and Culture Conditions

The *C. elegans* that were used to perform detection assays had the following features: strains N2, Bristol (wild-type) and CB1370, *daf*-2 (e1370). They were provided by the Caenorhabditis Genetics Centre at the University of Minnesota. All worms were age-synchronized and pipetted onto Nematode Growth Medium (NGM) in 55 mm Petri plates, where the temperature was kept at 20 °C.

The worms were fed with strain OP50 of *Escherichia coli*, seeded in the middle of the plate to avoid occluded wall zones, as *C. elegans* tend to stay on the lawn. To reduce fungal contamination, fungizone was added to the plates [88], and FUdR (0.2 mM) was added to the plates to decrease the probability of reproduction.

In the laboratory, the operator had to follow a specific methodology to obtain the images: (1) Extract the plates from the incubator to put them in the acquisition system; (2) Examine the lid, before initiating the capture, to check there is no condensation and wipe it if detected; (3) Capture a sequence of 30 images per plate at 1 fps and return the plates to the incubator. This method was designed to prevent condensation in the cover.

Image capture method

Images were captured using the monitoring system developed in [67], that uses the active backlight illumination method proposed in [64]. Active backlighting technique has been shown to be effective for low-resolution *C. elegans* applications for both the mentioned [64] and [68] capture systems. The monitoring system is based on an RGB Raspberry Pi camera v1.3 (OmniVision OV5647, with a resolution of 2592 × 1944 pixels, a pixel size of 1.4 x 1.4 µm, a view field of 53.50° × 41.41°, optical size of 1/4″, and focal ratio of 2.9) placed in front of the lighting system (a 7″ Raspberry Pi display 800 × 480 at a resolution at 60 fps, 24 bit RGB colour) and the inspected plate in between both devices. A Raspberry Pi 3 was used as a processor to control lighting. The distance between the camera and the Petri plate was sufficient to enable a complete picture of the Petri plate, and the camera lens was focused at this distance (about 77 mm). The system captured images of 1944x1944 pixels at a frequency of 1Hz. A graphic of the capture system is shown in Fig. 2.25.

The code, components and the guidelines to build the monitoring system and the assembly description can be found in the repository `https://github.com/JCPuchalt/SiViS`.

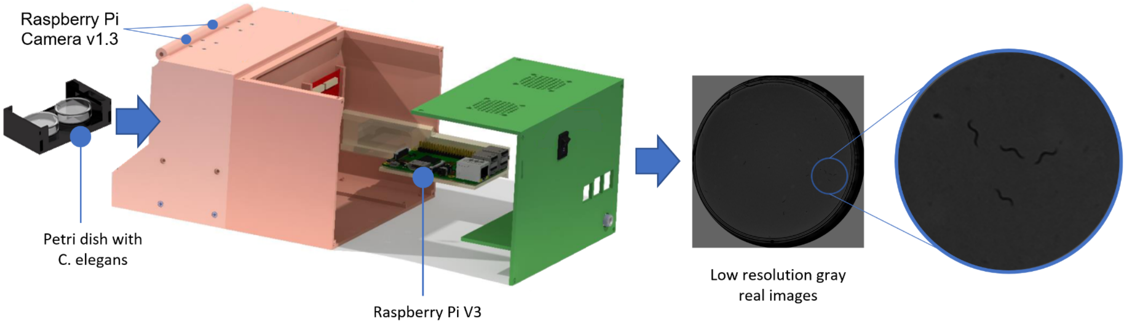


Figure 2.25: **Image capture system.** Location of the Petri dishes, as well as the other components of the image acquisition system.

Dataset generation and labeling

The dataset is composed of 1901 images captured from 106 test plates with the SiViS system [64]. Each plate contains between 10 and 15 nematodes. These images were manually labeled by marking the bounding boxes of each nematode.

The dataset was divided into three subsets using the following proportions: 70% for training, 10% for validation and 20% for test. Number of images and labels in each subset are shown Table 2.7.

Table 2.7: **Number of images and labels in each subset.**

	Images	Labels
Train	1330	14004
Val	190	1993
Test	381	3920

Detection method

In this article it was decided to test with the best-known detection methods based on artificial neural networks: YOLOv5 and Faster R-CNN.

Faster RCNN is an architecture proposed in [72]. It consists of three parts: (1) a convolutional neural network that extracts the image features; (2) a Region Proposal Network (RPN) in charge of obtaining the candidate bounding boxes and (3) a network in charge of performing the classification and regression of the bounding box coordinates.

YOLOv5 is an object detection algorithm created by Glenn Jocher in 2020. Considering previous models of the YOLO series, YOLOv5 offers higher accuracy, faster calculated speed and smaller size. Among the YOLOv5 models, YOLOv5s works the fastest, although the average precision is the lowest. The YOLOv5 network consists in three main parts: Backbone, Neck and Head. Backbone gathers and produces picture characteristics on various image granularities when the image is supplied. The picture characteristics are then stitched and transmitted to the prediction layer by Neck, where the image features are predicted by Head to generate bounding boxes and predicted categories [10].

Training and evaluation method

The networks were implemented and trained using the Pytorch deep learning framework [61] on a computer Gigabyte Technology Z390 AORUS PRO machine, Intel(R) Core (TM) i9-9900KF CPU @ 3.60GHz x16 with 32GB of RAM, and NVIDIA GeForce RTX 2080 Ti graphics card with 4352 Cuda cores, Ubuntu 19.04 64bits operating system.

Faster R-CNN. In this work the Faster R-CNN model architecture was used with a ResNet-50-FPN backbone pre-trained with COCO dataset [48] (Fig. 2.26).

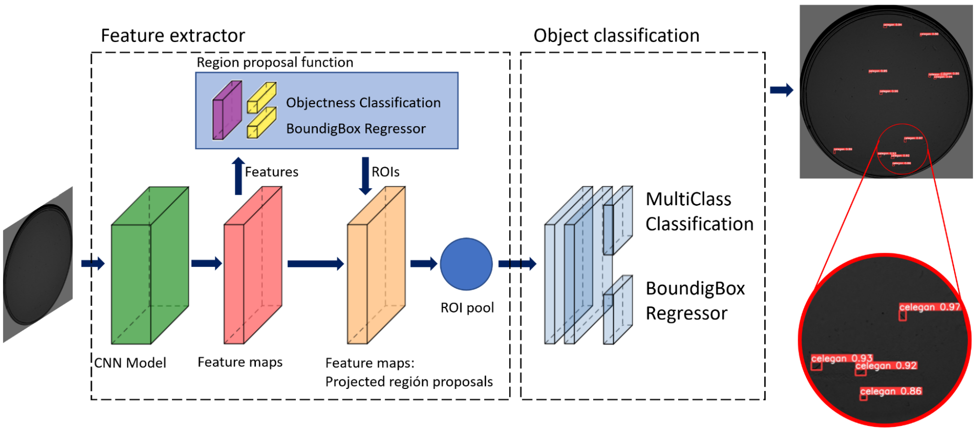


Figure 2.26: **Image pipeline through Fast R-CNN architecture.** The blocks of the Fast R-CNN architecture were made using the PlotNeuralNet tool [30].

The network was trained for 150 epochs with a batch size of 6 samples using the implementation of `https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html`. A Reduce Learning Rate on Plateau scheduler with a patience of 3 epochs and a factor of 0.1 was used. The optimizer used was SGD with momentum 0.9 and weight decay 0.0005. Vertical flip (0.5), HSV augmentation (hue=0.015, saturation=0.7, value=0.4), image scale (0.5) and translation (0.1) were used as data augmentation techniques.

YOLOv5. To train YOLOv5 we used the implementation of (`https://github.com/ultralytics/yolov5`) in its YOLOv5s version, which is the smallest and fastest (Fig. 2.27).

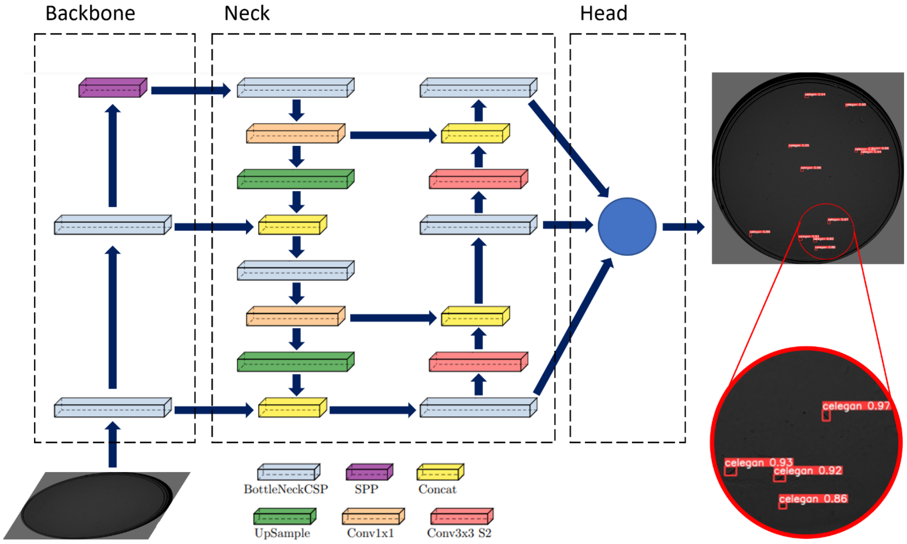


Figure 2.27: **Image pipeline through YOLOv5 architecture.** The blocks of the YOLOv5 architecture were made using the PlotNeuralNet tool [30].

We used the weights pretrained on the COCO dataset. It was trained with a batch size of 6 samples for 150 epochs using the SGD optimizer with momentum 0.937 and weight decay 0.0005. LambdaLR Learning Rate scheduler was used with the configuration proposed in the original implementation and other data augmentations, such as Vertical flip (0.5), HSV augmentation (hue=0.015, saturation=0.7, value=0.4), image scale (0.5) and translation (0.1) were used.

The YOLOv5 model needed an image size multiple of 32 and, consequently, this model resized automatically the images to 1952 pixels. To make both algorithms work in the same image conditions, the images were resized with the same scale factor for Faster R-CNN model. Before this, in this model's dataloader, a function to crop and resize the images was implemented, since Faster R-CNN model needed an image size between 800 and 1333 pixels.

Evaluation methods (Metrics)

To compare YOLOv5 and Faster R-CNN it is necessary to use some metrics that can be applied to both models, so the comparison is fair and adequate. The main metrics that will be used are computational cost and mean Average Precision (mAP), which is obtained from other metrics, precision and recall.

To decide whether a prediction is correct, the IoU is used, which measures the degree of overlap between the real box delimiting the object and the one predicted by the network. If the IoU value exceeds a threshold, it is considered True Positive (TP), otherwise it is considered False Positive (FP). If the network does not find an object, it is considered False Negative (FN).

Precision: It represents the relation between the correct detections and all the detections made by the model (Eq 2.16).

$$Precision = \frac{TP}{TP + FP} \tag{2.16}$$

Recall: It shows the ability of the model to make correct detections out of all ground true bounding boxes (Eq 2.17).

$$Recall = \frac{TP}{TP + FN} \tag{2.17}$$

Average Precision (AP): It is calculated as the area under the precision-recall curve, thus summarizing the information of the curve in one parameter (Eq 2.18).

$$\mathrm{AP} = \int_0^1 \mathrm{P(R)}d\mathrm{R} \tag{2.18}$$

Mean Average Precision (mAP): This is obtained calculating the mean of APs over all IoU tresholds. In this case, the range of IoU tresholds used is [0.5; 0.95] (Eq 2.19).

$$\mathrm{mAP} = \frac{1}{N}\sum \mathrm{AP}(n) \tag{2.19}$$

Where N is the total number of IoU thresholds and AP(n) is the average precision for each IoU threshold value.

Computational cost: It measures the amount of resources a neural network uses in a process, which determines the amount of time the process lasts. In this case, time per prediction will be measured to compare the computational costs in inference, which allows to determine how both models would work for possible real-time applications. For this reason, this metric will be measured in seconds.

2.3.4 Experimental results

This section presents the results obtained after training with the YOLOv5s and Faster R-CNN networks. The results were obtained with the best trained models in the three subsets (train, validation, test, Table 2.8, 2.9, 2.10 respectively).

Table 2.8: **Train results.**

Model	Parameters	mAP@0.5	mAP@[0.5-0.95]
YOLOv5s	7.5M	0.961	0.762
Faster R-CNN	41.3M	0.983	0.759

In Figures 2.28, 2.29, 2.30 we can find examples of true positives, false positives and false negatives respectively obtained when predicting with the models.

Table 2.9: **Eval results.**

Model	mAP@0.5	mAP@[0.5-0.95]
YOLOv5s	0.957	0.776
Faster R-CNN	0.947	0.705

Table 2.10: **Test results.**

Model	time/prediction (ms)	mAP@0.5	mAP@[0.5-0.95]
YOLOv5s	6.03	0.932	0.806
Faster R-CNN	43.25	0.944	0.691

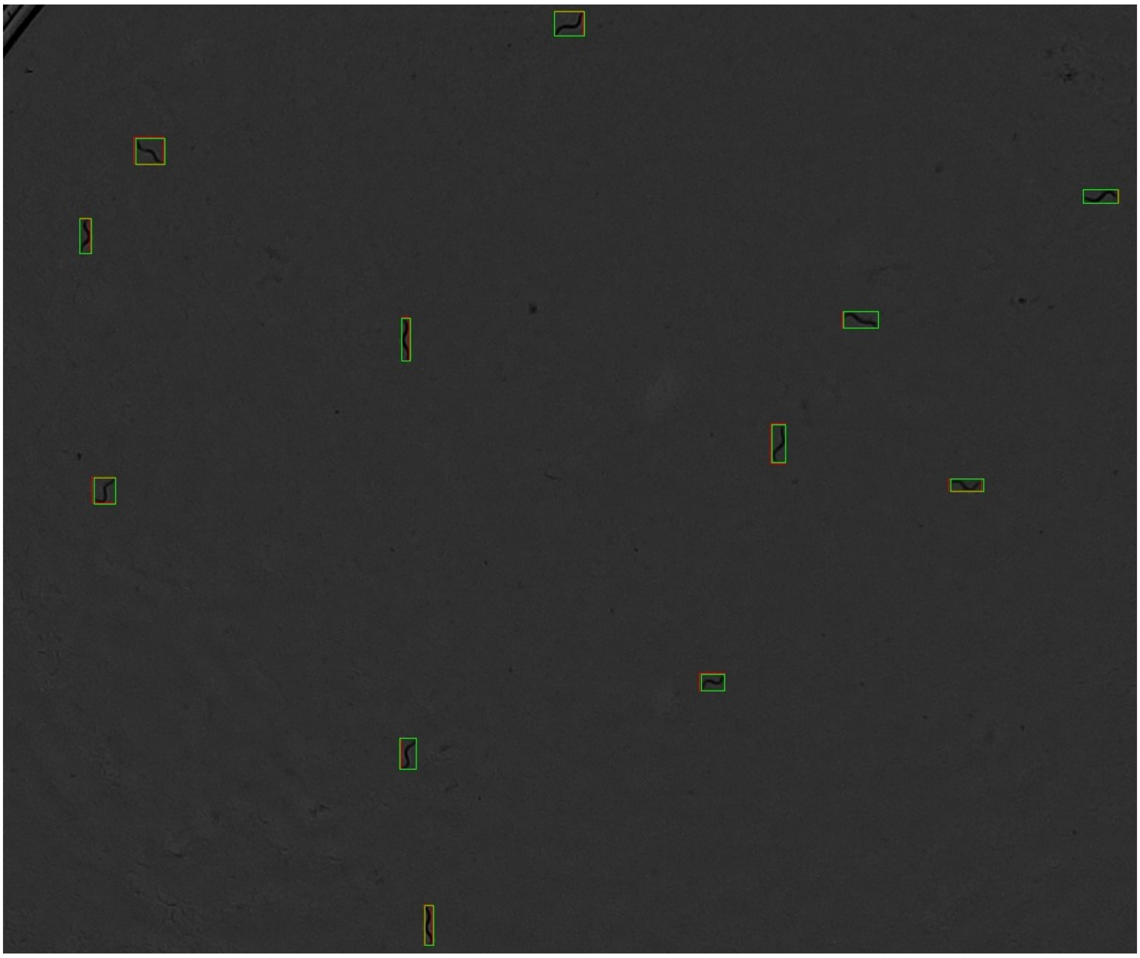


Figure 2.28: **Example of true positives obtained by applying the detection models.** The model predictions are shown in red and the labels in green.

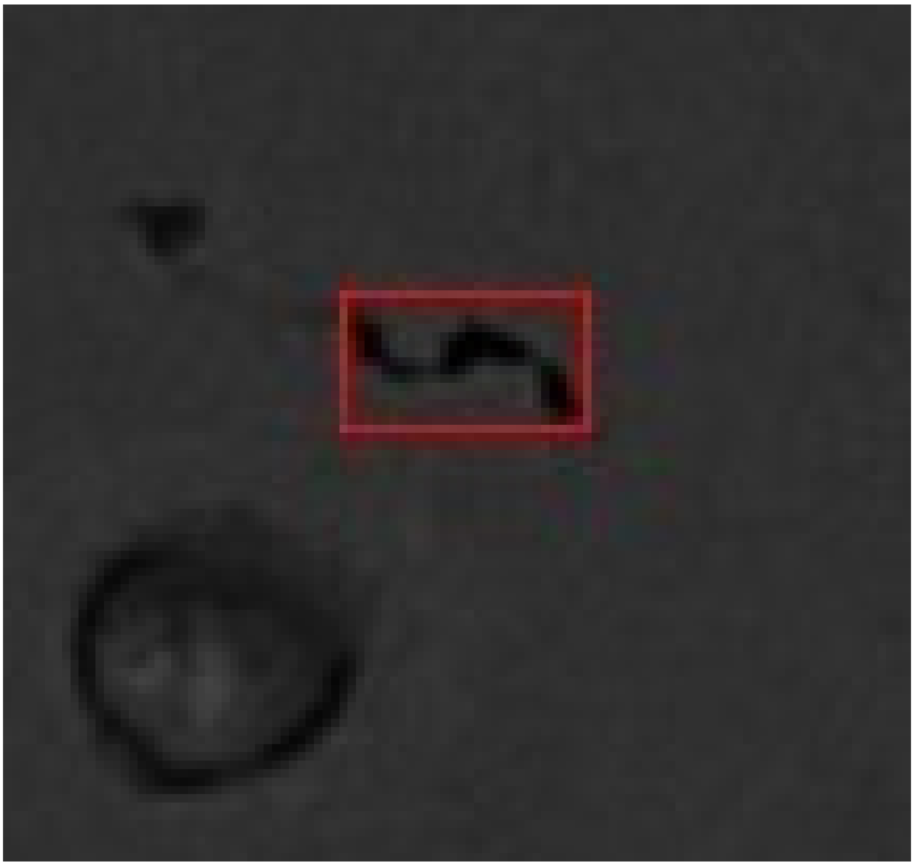


Figure 2.29: **Example of false positives due to dirt of the plate.** The model predictions are shown in red and the labels in green.

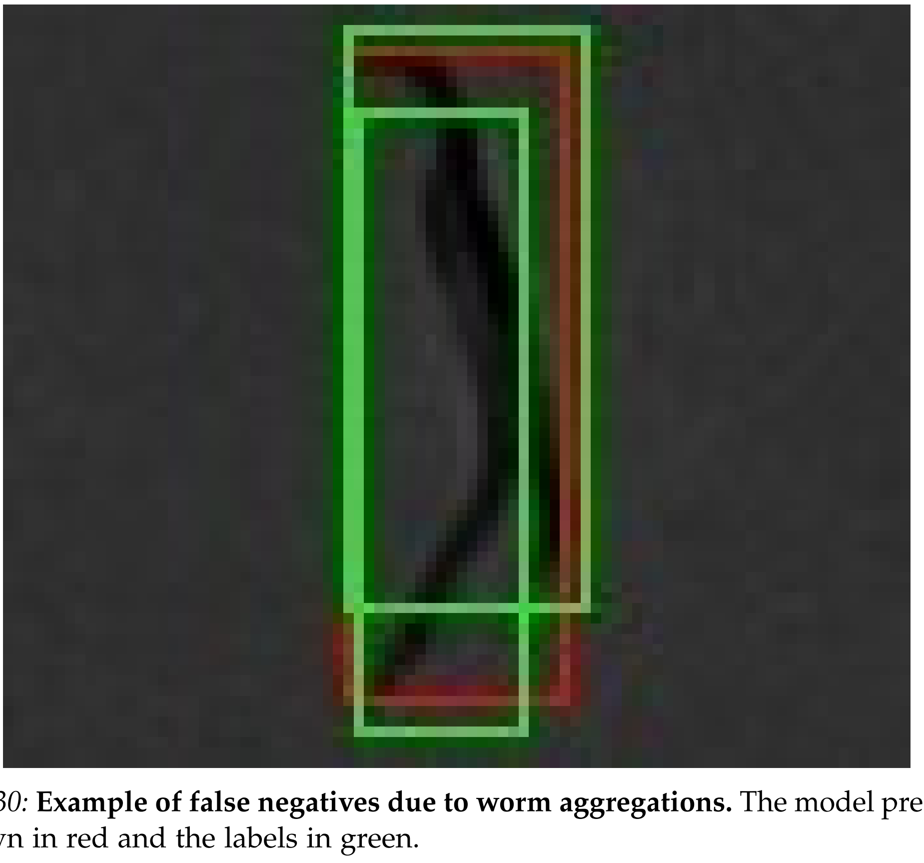


Figure 2.30: **Example of false negatives due to worm aggregations.** The model predictions are shown in red and the labels in green.

2.3.5 Discussion

The experiments carried out in this study have shown that, as expected, due to the greater number of parameters of the Faster R-CNN model used (backbone resnet50) with respect to YOLOv5s, the computational cost is higher, as reflected in the inference times of both models. Generally, in *C. elegans* image acquisition, the images are captured at a frequency of 1 Hz, maximum of 15 Hz, due to the low speed of movement of the nematodes. It has thus been shown that both detection techniques could be used for real-time monitoring of *C. elegans*. This would be useful for healthspan assays in which the movements of the nematodes are analysed.

In terms of network performance, the results of the experiments show that they perform similarly, achieving a mAP value of around 93-94% for an IoU treshold of 0.5. As explained in section *2.6 Evaluation methods (Metrics)*, this metric is quite representative, as it considers precision and recall values. The YOLOv5s model obtains a better mAP result for IoU treshold [0.5-0.95].

2.3.6 Conclusion and future work

Finally, it can be concluded that the YOLOv5s model performs better in this application of *C. elegans* detection, as it achieves similar accuracy results to those obtained with Faster R-CNN (accuracy values of 93.2% and 94.4% respectively for our dataset) but in less time and, consequently, with a lower computational cost.

As future work, the available dataset could be labelled by obtaining the nematode masks. With this new dataset, a Mask R-CNN network could be trained, which would have the advantage over YOLO of obtaining a segmentation. Another option to avoid the cost of labelling would be to use synthetic images to train the network.

Acknowledgments

This study was supported by the Plan Nacional de I+D with Project RTI2018-094312-B-I00, FPI Predoctoral contract PRE2019-088214, Ministerio de Universidades (Spain) under grantFPU20/02639 and by European FEDER funds. ADM Nutrition, Biopolis SL, and Archer Daniels Midland provided support in the supply of *C. elegans*.

2.4 Paper4

Skeletonizing *Caenorhabditis elegans* based on U-Net architectures trained with a multi-worm low-resolution synthetic dataset

Pablo E. Layana Castro[1], **Antonio García Garví** [1], **Francisco Navarro Moya**[1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Layana Castro, P. E., García Garví, A., Navarro Moya, F., Sánchez-Salmerón, A. J. (2023). Skeletonizing *Caenorhabditis elegans* based on U-Net architectures trained with a multi-worm low-resolution synthetic dataset. International Journal of Computer Vision. https://doi.org/10.1007/s11263-023-01818-6.

2.4.1 Abstract

Skeletonization algorithms are used as basic methods to solve tracking problems, pose estimation, or predict animal group behavior. Traditional skeletonization techniques, based on image processing algorithms, are very sensitive to the shapes of the connected components in the initial segmented image, especially when these are low-resolution images. Currently, neural networks are an alternative providing more robust results in the presence of image-based noise. However, training a deep neural network requires a very large and balanced dataset, which is sometimes too expensive or impossible to obtain. This work proposes a new training method based on a custom-generated dataset with a synthetic image simulator. This training method was applied to different U-Net neural networks architectures to solve the problem of skeletonization using low-resolution images of multiple *Caenorhabditis elegans* contained in Petri dishes measuring 55 mm in diameter. These U-Net models had only been trained and validated with a synthetic image; however, they were successfully tested with a dataset of real images. All the U-Net models presented a good generalization of the real dataset, endorsing the proposed learning method, and also gave good skeletonization results in the presence of image-based noise. The best U-Net model presented a significant improvement of 3.32% with respect to previous work using traditional image processing techniques.

keyword: Synthetic dataset, low-resolution image, U-Net, skeletonizing, end points, *Caenorhabditis elegans.*

2.4.2 Introduction

Extracting the central line or skeleton of a worm from images is not an easy task, and much less so in low-resolution images when there are aggregations between worms or the worms aggregate with plate noise. We use the word "noise" to refer to dark objects or segmentations, residues, stains or worm shapes, which are not actually worms. All these cases cause classical skeletonization algorithms to fail, leading to erroneous results. As demonstrated in [41], compared to classical skeletonization techniques, improved techniques can better locate and identify worms in the aforementioned cases, facilitating the automation of monitoring tasks, posture recognition, behavioral studies, etc.

Over the years, many applications have been developed for automatic monitoring and inspection of *C. elegans* using classical image processing techniques. Many of these applications solve this problem by identifying the central line or skeleton of the *C. elegans*. Skeleton identification basically reduces the shape of the worms without losing information about their posture. In order to identify the worms within images, some methods have been implemented to extract characteristics of the worm from the skeleton. The best-known characteristics include endpoints [22], smoothness [77, 76], length [101], previous segmentation [95, 100], or a combination of these [42]. Other applications for extracting worm skeletons use neural network techniques [8, 47, 25].

Methods that use neural networks are becoming more reliable and precise, helping many professionals and researchers to achieve their goals in many fields of science. However, a problem implicit to using these techniques is having a dataset large enough to train and validate the model, in addition to testing the results. Moreover, creating a labeled dataset is usually a time-consuming manual task. In this context, some neural network applications have proven that the use of synthetic data can solve this problem either partially (mixing real and synthetic data) or completely (purely synthetic data).

Generally, in *C. elegans* assays, these applications are performed using high-resolution images where there is only one worm or very few worms per plate. The advantage is that better processing results are obtained due to the number of pixels and that overlaps or aggregations between worms are avoided. A recent work [25] proposes a simulator to generate purely synthetic high-resolution images using reverse skeletonization, this technique employs small rectangular image patches to generate worm images.

We present a new convolutional neural network skeletonization method trained using purely synthetic low-resolution images. The convolutional network architecture used is the U-Net [63]. We take advantage of this U-shaped architecture with an encoder and decoder to produce encoded images of worm skeletons from low-resolution grayscale images. Instead of generating new synthetic frames of individual poses with rollings and self-intersections as in [25], our simulator generates new synthetic frames of multi-worm poses with intersection behaviors and parallel contacts. The results show a significant improvement of 3.32% compared to a previous work [41] which improved classical skeletonization methods.

Highlights:

- A *C. elegans* skeletonization method is proposed based on U-Net type neural networks with low-resolution images and noise.
- A new method for generating low-resolution synthetic images is proposed to easily generate a custom-labeled dataset for different *C. elegans* behaviors.
- A neural network has been trained with a low-resolution synthetic image and successfully tested in the domain of real images.
- Different U-Net architectures have been compared with an algorithm based on traditional image processing techniques.

2.4.3 Related Work

In this section, we review the state of the art of various works related to different network architectures and neural network techniques applied to *C. elegans*.

Caenorhabditis elegans and neural networks

Caenorhabditis elegans is one of the most widely studied organisms and has acquired great importance in the field of biology [5]. Its genome has been annotated in great detail, and research shows that many human diseases have homologues in the genome of this nematode [58], making it an attractive animal model for the study of human pathologies. The advantages offered by this organism with respect to others include its short life cycle (around 21 days), short reproductive period, small size (around 1mm long), and feeding based on bacterial strains such as *Escherichia coli*; all of which facilitate its large-scale culture [11].

In the past, *C. elegans* assays were monitored manually but nowadays many researchers choose more automatic technologies, thus reducing both processing times and the fatigue of technicians, who would otherwise spend hours looking through the microscope daily. This is where computer-vision applications have a great advantage over these manual practices. Due to the flexible body and the different poses that *C. elegans* can adopt, many automatic applications use skeletonization techniques to solve problems related to healthspan [23, 44, 14], lifespan [35, 40, 65], tracking [33, 38], behavior monitoring [102, 62], etc. These techniques are very useful in microscopic images without noise, but low-resolution and noisy images present a challenge difficult to overcome. Other automatic methods use neural networks (NN), which are more robust against these problems.

Neural networks have different topologies and these can perform tasks of classification, segmentation, detection, and so on. All these architectures can automatically extract features, with which *C. elegans* applications are developed, such as skeletonization [25], extreme segmentation [52], and others [98, 96, 97, 103]. Furthermore, networks that have an encoder and a decoder have proven capable of solving more complex tasks such as posture classification [34], skeleton definition in microscope images [8] or patch acquisition to resolve aggregation [51].

U-Nets

The U-Net is a neural network with an encoder and a decoder. Since it first appeared, the U-Net [79] has been widely used not only in medical applications, for which it was first introduced, but also in the segmentation of animals [24, 59], objects [104, 99], etc. Various authors have taken this network architecture as a reference and have modified it to achieve greater convergence in training. These modifications consist of adding, removing, or replacing convolutional layers with others, thereby increasing or decreasing the size of the network. However, making the network deeper (increasing the number of convolution layers) or increasing the width of the network does not always result in a better prediction, it all depends on our dataset.

SmaAt-UNet [93] proposes reducing the size of the base U-Net [79] by adding spatial-channel attention, and shows that it can achieve similar precision values, this leads to reducing the inference time or the resources needed during the exploitation phase. On the other hand, UMF U-Net [63] modifies the U-Net standard by adding BatchNorm and dropOut layers and shows that careful choice of hyperparameters and other training configurations play a very important role in network development. As well as these cases, there are other more recent modifications [1, 55, 91, 94, 46, 49, 69, 2, 54, 27, 7, 31] that present significant improvements in the precision of their respective datasets compared to the standard U-Net.

When working with neural networks, it is problematic to obtain a well-labeled and balanced dataset. This can be a very expensive or even an impossible task; therefore, to alleviate this problem, different data augmentation methods have been developed, as well as simulators for generating synthetic images.

Data augmentation and synthetic images

The use of synthetic data can help attain network convergence, avoid overfitting, and improve data generalization. On the one hand, the mixture of synthetic data with real data has been shown to help improve the training of neural networks [60, 15, 3, 13], even in applications with *C. elegans* [21]. In general, these types of techniques convert synthetic images to the domain of real images, in order to achieve similar distributions. This domain change is achieved thanks to architectures such as GAN [47], encoders, and decoders [9].

On the other hand, applications that only use synthetic images also provide good results in the domain of real images, [82, 26, 53]. The use of simulators to create purely synthetic images pave the way to being able to create larger and more variable datasets, thus being able to generate cases or events that occur infrequently in real images. The simulation of *C. elegans* in low-resolution images is a challenge that, to our knowledge, has not been addressed before. Accordingly, [25] uses a high-resolution synthetic image simulator of a single worm to train a neural network.

In this work, we aim to generate low-resolution synthetic images to train a segmentation neural network of worm skeletons from 20 to 40 points. Our method obtains outstanding results, outperforming previous skeletonization work that improved classical image processing techniques.

2.4.4 Methods

Strain and culture conditions of *C. elegans*

The *C. elegans* used were wild type (N2) and CB1370, *daf*-2(e1370) young adults, obtained from worm eggs incubated at 20ºC in 55mm diameter NGM plates at the Caenorhabditis Genetics Center (CGC), University of Minnesota. *Escherichia coli* strain OP50 was used as food. To prevent reproduction, FUdR (0.2 mM) was used, and to prevent contamination by fungi, fungizone (1µg/mL) was added and the plates were closed with a lid. The standard method [88] was followed to remove those plates with contamination. Plates with 10, 15, 30, 60, and 90 nematodes were cultured to obtain greater variability and analyze different types of behavior (aggregation of two or more *C. elegans*, aggregations with plate noise and occlusions).

Real-image acquisition method

Images of complete 55 mm diameter Petri dishes were captured. Image acquisition was performed by a laboratory operator at a temperature of 20ºC using the hardware and software described in [67]. To capture the images, the laboratory operator removed the plates from the incubator and placed them in the capture system (Figure 2.31) where the system proceeded to capture a sequence of images of 1944x1944 pixels at a frequency of 1Hz. The *Escherichia coli (E. coli)* OP50 strain was seeded in the center of the plate to prevent *C. elegans* from moving out of the field of view, either by climbing up the edges of the plates or by positioning themselves near the plate edges. Those plates with condensation on the cover were withdrawn from the image acquisition process.

The abovementioned system [67] was developed with open hardware and software, using a Raspberry Pi v1.3 RGB camera, OmniVision OV5647 with resolution of 2592 × 1944 pixels, and pixel size of 1.4 × 1.4 µm, field of view of 53.50° × 41.41°, with 1/4″ optical size and 2.9 focal ratio, a lighting system based on a 7″ Raspberry Pi screen with a resolution of 800 × 480 at 60 fps, 24-bit RGB color and as processing unit a Raspberry Pi 3, (Figure 2.31). The mounting process and image capture are detailed in [67].

The lighting technique used was active backlighting. This technique has been shown effective for low-resolution *C. elegans* applications for both the aforementioned [64] and [68] capture systems. Active backlighting consists of reducing the variability of the captured images by keeping gray scales more constant. This made it possible to differentiate the background of *C. elegans* easily in all images. To capture the real image sequences, our object of interest (Petri dish with worms) was placed between the illumination system and the camera, as described in [67].

With this configuration the nematodes have a maximum size of 40x4 pixels and a minimum of 20x3 pixels. In reality, the worms measure 1 mm. Working with low resolutions may complicate the problem in some cases, but it has advantages in terms of computational and memory efficiency. This resolution is sufficient to automate assays such as lifespan, healthspan, etc.

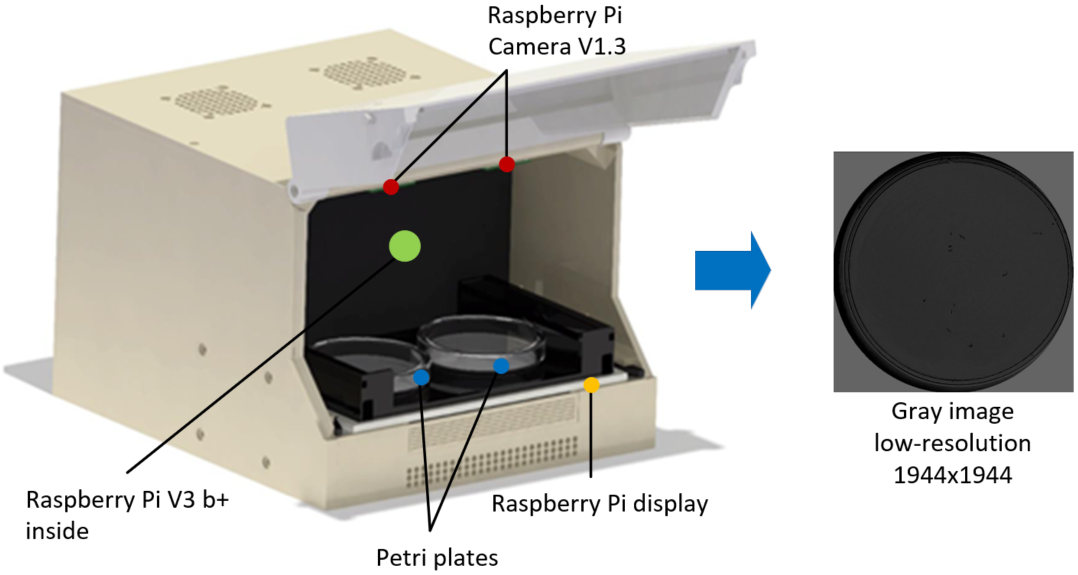


Figure 2.31: **Image capture system.** Location of the Petri dishes, as well as the other parts of the capture system [67]

Image Simulation Method

A simulator has been designed capable of generating new sequences both of aggregation behaviors between worms (parallel behaviors and intersections) and of augmenting individual behaviors (free motion, coiling) with geometric transformations. Figure 2.32 shows a conceptual outline of the synthetic image generation process.

The simulator starts from a database containing manually labeled real *C. elegans* paths and images of Petri dishes without worms. The labels contain the location (X,Y) of each of the points of the skeleton, as well as the color and width of each point of the skeleton.

The process to simulate a sequence consists of selecting K paths randomly, applying rotation and translation transformations to them, and combining them to obtain the desired behaviors, thus obtaining an integrated sequence. To generate the parallel aggregation behavior, one path is randomly selected and a new path is generated by modifying its color and XY position. The XY position of the first worm is at NR, while the second worm is at NR + W1. W1 is the smallest width of the worm body (1 or 2 pixels). The simulator generates two types of parallel behaviors, in the first, both worms navigate in the same direction, while in the second one of them navigates in the opposite direction. To achieve the second case the path of one of the two worms is rotated 180º, i.e., the worm of the rotated path starts at the end of the path and navigates in parallel with the other worm approximately in the middle of the path.

To generate the intersection aggregation behavior, two paths are randomly selected, the intersection point will be a random point on the skeleton of each worm from a random pose of each path. Each path has 30 poses and each worm can have between 20 and 40 skeleton points.

We have observed in the real dataset that aggregation behaviors (intersection and parallel) are accompanied by speed changes and pauses, so we randomly added these interactions within the simulator. Pauses are simulated by repeating poses for two instants of time. Speed changes are obtained by skipping a pose in the path.

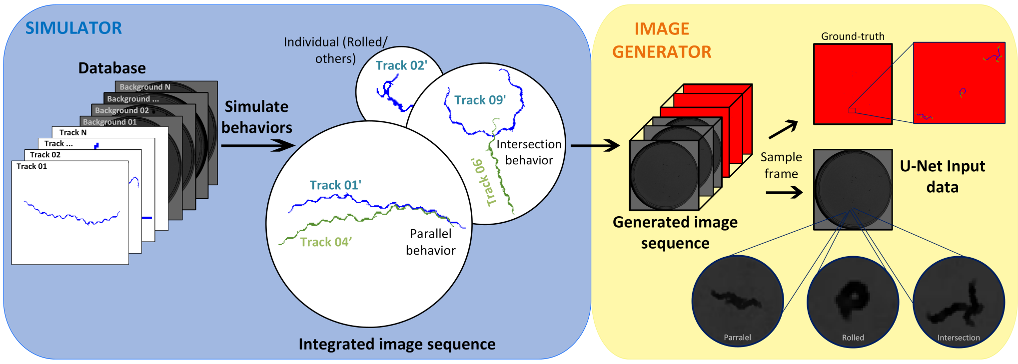


Figure 2.32: **Conceptual outline of the synthetic image generation process.**

The trajectory simulation consisted of applying a rotation angle (θ) to all the skeletons of a worm sequence and moving that sequence to a random X-Y point inside the plate. The values of N, θ and CC the centroid of the Petri dishes (pre-process info) were needed to calculate the new X-Y position of the worm trajectory (Equation 2.21, Equation 2.22), Figure 2.33a, b. The angle θ and NR were randomly generated between $0 - 2\pi$ and $0 - R$, respectively. The value of R was obtained from the difference between the plate radius (P) and the diameter of the trajectory found in the pre-process info file (2T), Equation 2.20.

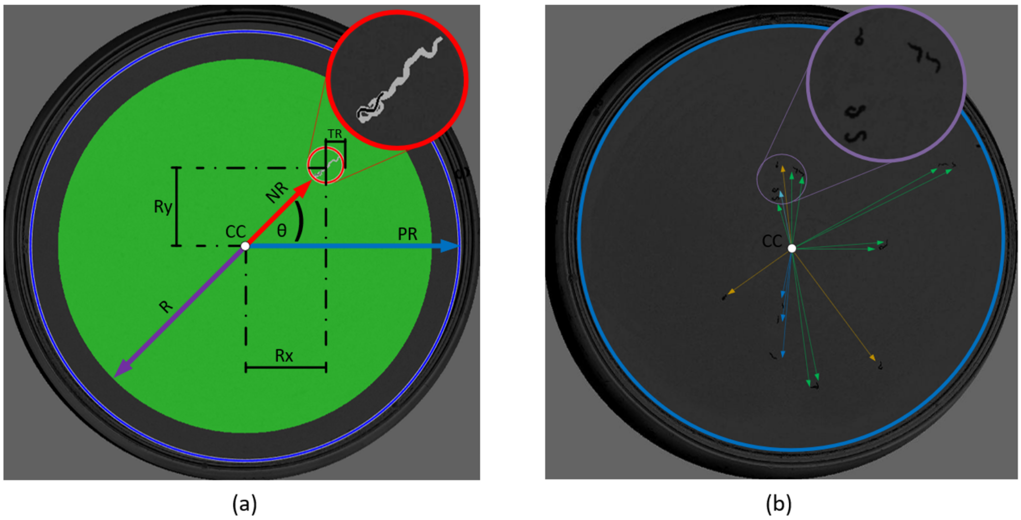


Figure 2.33: **Development of synthetic images. a)** Random generation of the track of a worm. **b)** Synthetic image with 16 worms on the plate.

$$R = P - 2T \tag{2.20}$$

$$N_x = CC_x + N\cos\theta \tag{2.21}$$

$$N_y = CC_y + N\sin\theta \tag{2.22}$$

To rotate the worm skeletons, a random angle (α) between $0 - 2\pi$ was generated. The rotation operation was performed by multiplying each skeleton pixel (Equation 2.24) by a rotation matrix (Equation 2.23):

$$M_a = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \tag{2.23}$$

$$P_i = \begin{bmatrix} P_x \\ P_y \end{bmatrix} \tag{2.24}$$

Then, using the integrated sequence, the image sequence is generated by drawing the paths on an empty Petri dish image. The paths are drawn by inserting circles of diameter equal to the width value of the skeleton point stored in the database into each of the pixels of the skeletons. To color the circle, the value of the skeleton point is used and also averaged with the background. Finally, a blur filter (3x3) is applied to the images. This filter was essential to bring the synthetic image domain closer to the real image domain. In addition, it favored convergence in the training of the networks. In this step, in addition to the gray image sequence, ground-truth masks are also generated.

This simulator, which has been designed to obtain sequences of behaviors, allows an image to be selected from the generated sequence as input to the network. The details of the code implementation have been added to Appendix A (A.3.1).

Classical skeletonization method

Classical skeletonization techniques have proven easy-to-implement for extracting shapes and predicting worm behaviors, which are problematic when worms coil or aggregate with each other, or with plate noise. When this happens, part of the skeleton is absent or displaced, and this is because these skeletonization techniques reduce the segmentation pixels until they achieve the central line. In these cases, skeleton prediction errors occur, as shown in Figure 2.34. If the aggregation occurs in a large part of the worm body it can lead to a large error.

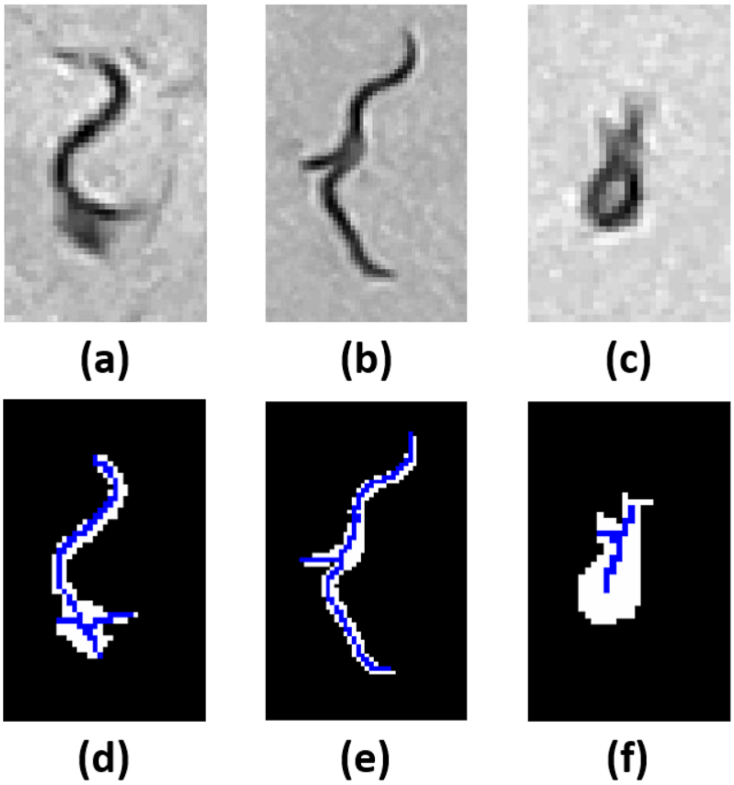


Figure 2.34: **Classical skeletonization of problematic cases. (a)** Grayscale image of worm aggregated with noise. **(b)** Grayscale image of two worms aggregated at one end and part of the body. **(c)** Gray image of worm coiled upon itself. **(d, e, f)** Result of classical skeletonization of images a, b, c, respectively. The white pixels show the segmentation using a threshold of 35, while the blue pixels show the result of skeletonizing that segmentation.

Skeletonization method using improved skeleton

This method involves obtaining improved skeletons (Figure 2.35a, b, c) from the width and length characteristics. These characteristics are obtained when the worms are free and not coiled during a previous preprocessing [41]. The advantage of this technique, unlike classical skeletonization techniques, is that it can separate connected or coiled worms through new skeletons. In general, other applications cancel tracks where aggregation between worms, aggregation with noise, and coiling occur; however, the improved technique is very useful in these cases, recognizing skeletons (poses) and predicting behaviors. [41] showing that an improved skeleton together with worm-specific features such as color and temporal image features can solve problems of aggregation between worms or with noise in image sequences.

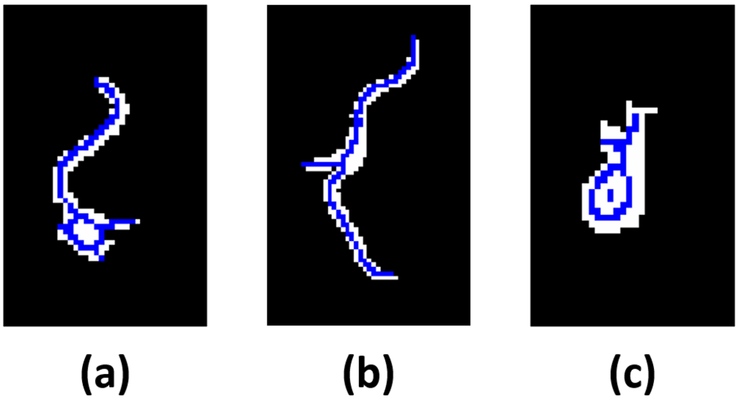


Figure 2.35: **Skeletonization with enhanced algorithm (ISA)** [41]. **(a, b, c)** Skeletonization result with an improved algorithm (ISA) of the gray images from Figure 2.34a, b, c, respectively. The white pixels show the segmentation using a threshold of 35, while the blue pixels show the result of skeletonizing that segmentation.

Proposed skeletonization method

The model used for the segmentation of worm skeletons was the convolutional neural network U-Net. Different U-Net architectures were analysed to compare their performance. Comparison was made of the models U-Net standard [79], Alexandre's U-Net [1], SmaAt-UNet, U-Net with DSC, U-Net with CBAM, U-Net with DSC, CBAM [93], UMF U-Net [63] and all showed good results.

Figure 2.36 shows all the blocks used with the different architectures. For the standard U-Net [79], the Doubleconv block does not have the white BatchNorm2d block and the yellow CBAM blocks and the purple Down4 block does not have the Dropout blocks. Alexandre's U-Net [1] is the same as the standard U-Net but includes the BatchNorm2d block inside the Doubleconv block. UMF U-Net [63] is the same as Alexandre's U-Net but on the purple Down4 block it does have the Dropout blocks. SmaAt-UNet [93] on the other hand has the spatial-channel attention blocks, it also has no dropout blocks in the purple Down4 block, and the models with DSC instead of the Doubleconv blocks have depthwise-separable convolutions blocks.

These models predict three classes: background, worm ends, and worm body. The background class (red pixels in Figure 2.36) are all those pixels that do not correspond to the worm skeletons, such as the plate edge and interior of the Petri dish, dark spots, residues inside the dish, etc. The worm-ends class (green pixels in Figure 2.36) includes those pixels corresponding to the head and tail of the worm skeleton. These vary between 5 and 10 pixels at each end, they are generally lighter pixels (higher grayscale intensity). And finally, the worm-body class (blue pixels in Figure 2.36) are pixels in the center of the skeleton and are darker than the worm-end pixels (less grayscale intensity).

Given the dimensions of the input image and the limitations of the hardware to train and validate the architecture, both the input image and the ground-truth image were divided into 4 equal parts. To reconstruct the prediction image, each output prediction part of the network was joined in the same order as the input image, as shown in Figure 2.36.

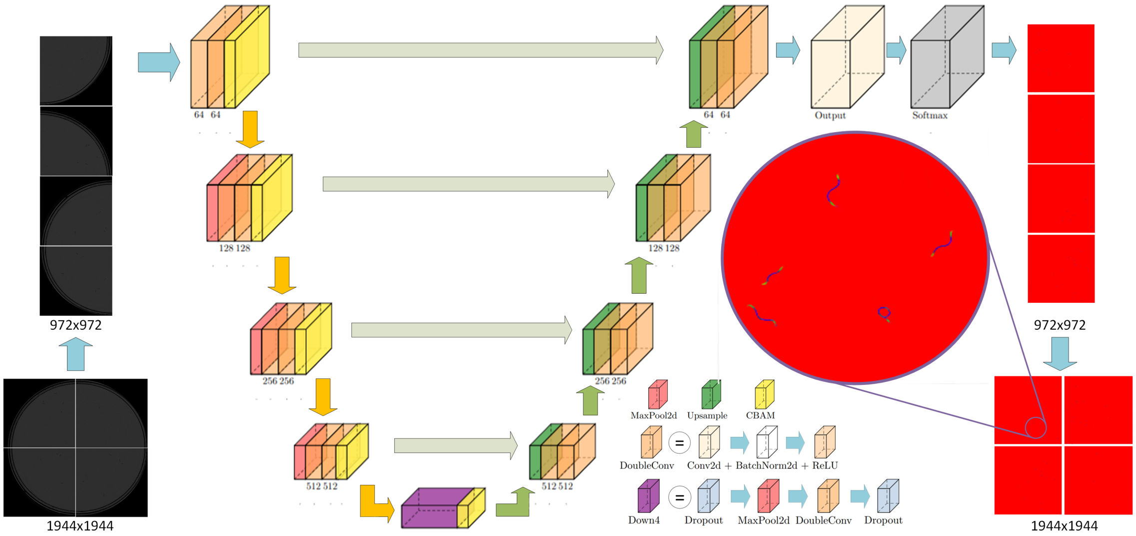


Figure 2.36: **Image pipeline through U-Net architecture.** The blocks of the U-Net architecture were made using the PlotNeuralNet tool [30]. The image is divided into 4 parts, each part enters the network and the result is reassembled to form a single image.

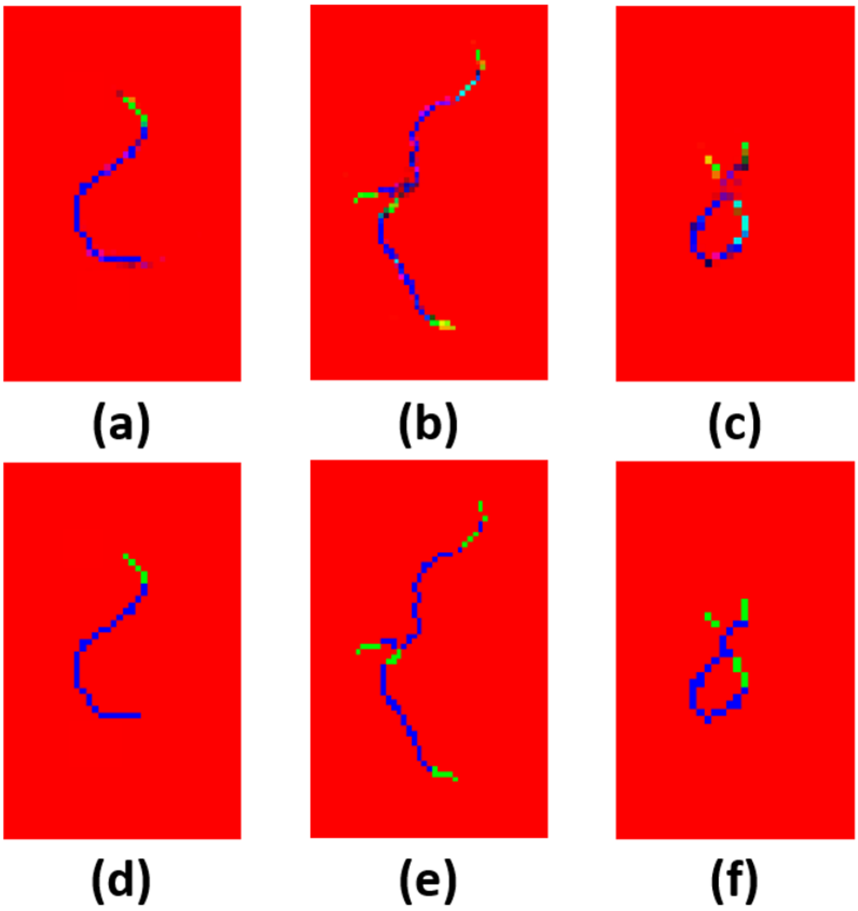


Figure 2.37: **Coding of the output image from the network. (a, b, c)** Resulting skeletons using the UMF U-Net [63], network from the gray images in Figure 2.34a, b, c respectively. **(d, e, f)** Pixel encoding using the maximum value of the RGB channels. Red pixels are background pixels, blue pixels are worm body pixels, and green pixels are worm-end pixels. The results obtained with the rest of the models are similar to these.

Evaluation method

To evaluate all the U-Net models, two datasets were used, a synthetic dataset that was used in the training and validation stages of the networks, and a dataset of real images to test the results in the cases of coiling, aggregation between worms, and aggregation with noise (Figure 2.38).

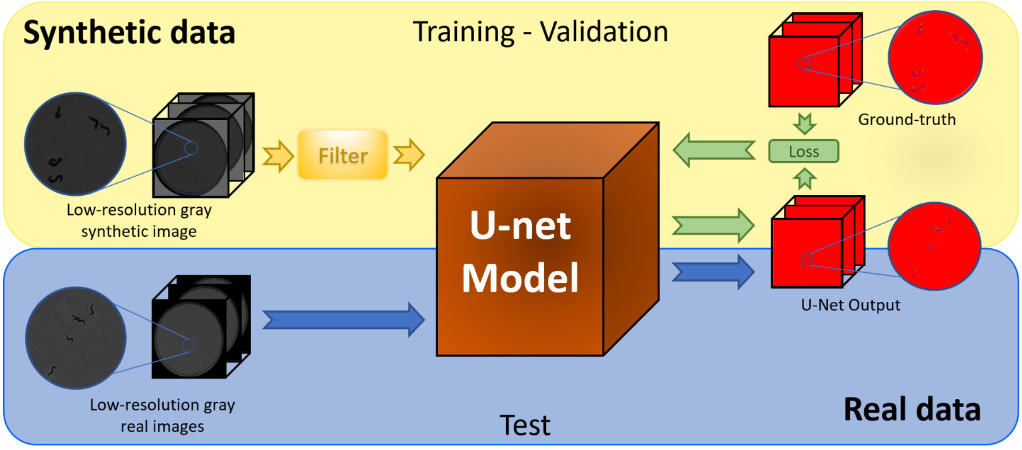


Figure 2.38: **Synthetic and real dataset pipeline.** The synthetic dataset was used to train and validate the U-Net neural network. The trained network was used to test the real image domain.

To evaluate the real dataset, the Jaccard index, also known as intersection over union (IoU), and euclidean distance were used. The IoU coefficient measures the accuracy of a prediction with respect to a ground-truth [39]. And as its name indicates, it is obtained by dividing the total area of the intersection by the union of these areas, Equation 2.25. On the other hand, the euclidean distance measures the average error in pixels of a prediction with respect to a ground-truth, Equation 2.26

$$IoU = \frac{\sum P_{w1} \bigcap P_{w2}}{\sum P_{w1} \bigcup P_{w2}} \tag{2.25}$$

$$E.D. = \sum_{i=1}^{nw} \frac{\sqrt{(X1_i - X2_i)^2 + (Y1_i - Y2_i)^2}}{nw} \tag{2.26}$$

The Precision and Recall metrics were used to evaluate the results of the detection experiment, and the MOTA metric was used to evaluate the results of tracking experiment. To obtain Precision (Equation 2.27) and Recall (Equation 2.28) metrics, three parameters were used: TP (true positives), FP (false positives), FN (false negative). On the other hand, to obtain MOTA metric (Equation 2.29), the FN, FP, IDS and GT parameters were used. GT was the total number of worms in the aggregation, IDS value was increased by 1 when the body of a predicted worm overlapped more with another worm than with its respective GT. For the overlap, the IoU value and a threshold of 0.5 were considered.

$$Precision = \frac{TP}{TP + FP} \tag{2.27}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.28}$$

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDS_t}{\sum_t GT_t} \tag{2.29}$$

The synthetic dataset labels (ground-truth) were obtained automatically from the simulator, while the real dataset labels (ground-truth) were obtained by manually labeling worm skeletons. This task was performed using a pixel labeling application. The pixels of each worm skeleton were selected one by one until the skeleton was complete.

The predictions are not always exact for all the pixels, usually, one or more pixels are displaced with respect to the real label (ground-truth), thus obtaining low measurement errors and incorrect skeleton indicators. When the worm is 3 pixels in diameter, the skeleton pixel is the center pixel, but if the worm has an even number of pixels, it is impossible to select the center pixel, which may result in false errors between manual labeling and pixel predictions of the skeleton. To solve this problem and obtain a better measurement of results, we decided to use the worm body to obtain a more significant IoU value that would better reflect the prediction of the skeleton. To recover the shape and body of the worm, a dilation operation was performed on all the pixels of the skeleton with a disk of radius 2 (approximate diameter of the worm). This operation was performed for all manual labels. IoU and Euclidean distance metrics have been calculated for the following classes: worm ends, worm body and worm (fusion of body and ends).

2.4.5 Experiments and results

Method comparison

In this experiment, different U-Net architectures were compared to find the most accurate one for our case. In addition, it was compared with the results of a method based on traditional computer vision techniques.

As previously mentioned, a synthetic dataset was used to train and validate the networks and a real dataset to test all the results. For the synthetic dataset, 400 sequences of 30 images were simulated, giving a total of 12000 images, 70% was used to train the network (8400) and the other 30% was used for the validation stage (3600). Each image of the sequence had 16 worms per plate, in which different cases of aggregation were simulated.

The hardware used for training and validation of the different networks was a Gigabyte Technology Z390 AORUS PRO machine, Intel(R) Core (TM) i9-9900KF CPU @ 3.60GHz x16 with 32GB of RAM, and NVIDIA GeForce RTX 2080 Ti graphics card with 4352 Cuda cores, Ubuntu 19.04 64bits operating system. The implementation was carried out in a Python version 3.7.5 environment, using the Pytorch 1.18, OpenCV 4.5.4, and SWIG 3.2 libraries.

Different U-Net architectures were compared and the training for each of these took about 48 hours with the abovementioned hardware. The hyper-parameters used for the training and validation phase were Batch_size = 1, num_workers = 1, maximum epoch = 10. The optimizer used was ADAM with a learning rate = 0.0001, betas = [0.95, 0.999], eps = 1e-8, CrossEntropyLoss() as the loss function, and the ReduceLROnPlateau scheduler with hyper-parameters mode = ′min, and patience = 2.

All the U-Net architectures used were trained and evaluated using the same training and validation dataset. After each training, the model with the lowest loss value in the validation phase was selected to evaluate all the results. The average loss resulting from the loss function CrossEntropyLoss() and average IoU values for each model are shown in Table 2.11.

Table 2.11: **Synthetic dataset loss and IoU results.** Both columns show the average for the results of the 3600 evaluation images of the synthetic dataset. The loss function used was CrossEntropyLoss(), the IoU index, and the euclidean distance (E.D.) were the ones described in the evaluation method.

		Avg. worm-ends class		Avg. worm-body class		Total average worm class	
Model	loss	IoU	E.D.	IoU	E.D.	IoU ± IC 95%	E.D. ± IC 95%
U-Net	6.47E-06	0.8883	0.1875	0.9315	0.0747	0.9322±2.40E-04	0.1067±5.12E-04
U-Net A.	3.05E-04	0.8179	0.1262	0.8573	0.1269	0.8597±1.24E-03	0.1270±1.06E-03
UMF U-Net	**1.54E-06**	0.9317	0.0019	**0.9371**	**0.0006**	**0.9361±9.30E-04**	**0.0009±6.99E-05**
SmaAt DS	2.16E-05	0.9153	0.0241	0.9297	0.0184	0.9287±1.09E-03	0.0199±3.68E-04
SmaAt DS AT	1.60E-05	0.9191	0.0226	0.9305	0.0112	0.9295±1.03E-03	0.0141±3.38E-04
SmaAt DS AT 4C	1.23E-04	0.8120	0.1263	0.8625	0.1073	0.8615±1.07E-03	0.1123±7.15E-04
SmaAt AT	3.06E-06	**0.9324**	**0.0017**	0.9364	0.0011	0.9355±9.42E-03	0.0012±9.44E-05

For the real dataset, 4500 images of Petri dishes were analyzed and those difficult cases were selected in which the worms coiled on themselves, aggregated to each other, or presented noise from the dish, in 90, 157, and 417 images, respectively. In order to obtain all this variability, the images contained 10, 15, 30, 60, and 90 worms.

The output images of all U-Net models were encoded using the maximum RGB channel value. Once the coded images of prediction and ground-truth were obtained, the pixels of the worm body and the worm ends were joined to form a single skeleton, then the shape of the worm was recovered as indicated in the evaluation method and the precision result was obtained using the IoU index. Table 2.12 shows the total parameters of each model and the average of the results obtained from the evaluation with these models and for all the cases analyzed.

Appendix B(A.3.2) shows the metrics obtained for each of the problematic cases: aggregation between worms (Table A.3), aggregation with noise (Table A.4) and rolled cases (Table A.5).

Table 2.12: **Average IoU results of the actual dataset.** Average IoU values of problematic cases using the encoding of the maximum value of RGB channels.

		Avg. Aggregation		Avg. Agg. with noise		Avg. Rolled		Total average worm	
Model	N Parameters	IoU	E.D.	IoU	E.D.	IoU	E.D.	IoU ± CI 95%	E.D. ± CI 95%
ISA		0.7625	0.5659	0.6421	2.1726	**0.8122**	**0.5540**	0.6936±0.0125	1.6185±0.1517
U-Net	17.2576M	**0.7634**	**0.5551**	0.6510	0.9001	0.7600	0.6678	0.6923±0.0134	0.8092±0.0466
U-Net A.	17.2664M	0.6858	0.7822	0.6980	0.6712	0.7172	0.6089	0.6977±0.0105	0.6274±0.0277
UMF U-Net	17.2664M	0.6992	0.6090	0.7313	0.7259	0.7622	0.5686	**0.7279±0.0066**	**0.6097±0.0276**
SmaAt DS	3.9536M	0.6339	0.5737	0.7133	0.6562	0.7481	0.5905	0.6993±0.0086	0.6529±0.0323
SmaAt DS AT	4.0320M	0.5759	0.5870	0.6849	0.6505	0.7004	0.6150	0.6613±0.0107	0.6642±0.0347
SmaAt DS AT 4C	3.9986M	0.5713	0.7471	0.7193	0.7032	0.7497	0.6053	0.6884±0.0125	0.6240±0.0194
SmaAt AT	17.3447M	0.6767	0.8359	**0.7338**	**0.6459**	0.7610	0.6146	0.7240±0.0082	0.6476±0.0232

As shown, the results of the networks were also compared with the values obtained in a previous work (ISA) [41]. The average results showed that for the real dataset UMF U-Net is the best skeletonization method, showing a statistically significant difference with the previous work [41] Figure 2.40. It should be noted that for cases of aggregation with noise it is the best option.

Figure 2.39 compares the previous work with respect to the UMF U-Net architecture using a box plot. Figure 2.41 shows the results obtained in an image section of the different architectures used. Although the results are similar, Figure 2.41d of the UMF U-Net predicts more connected skeletons than the other architectures.

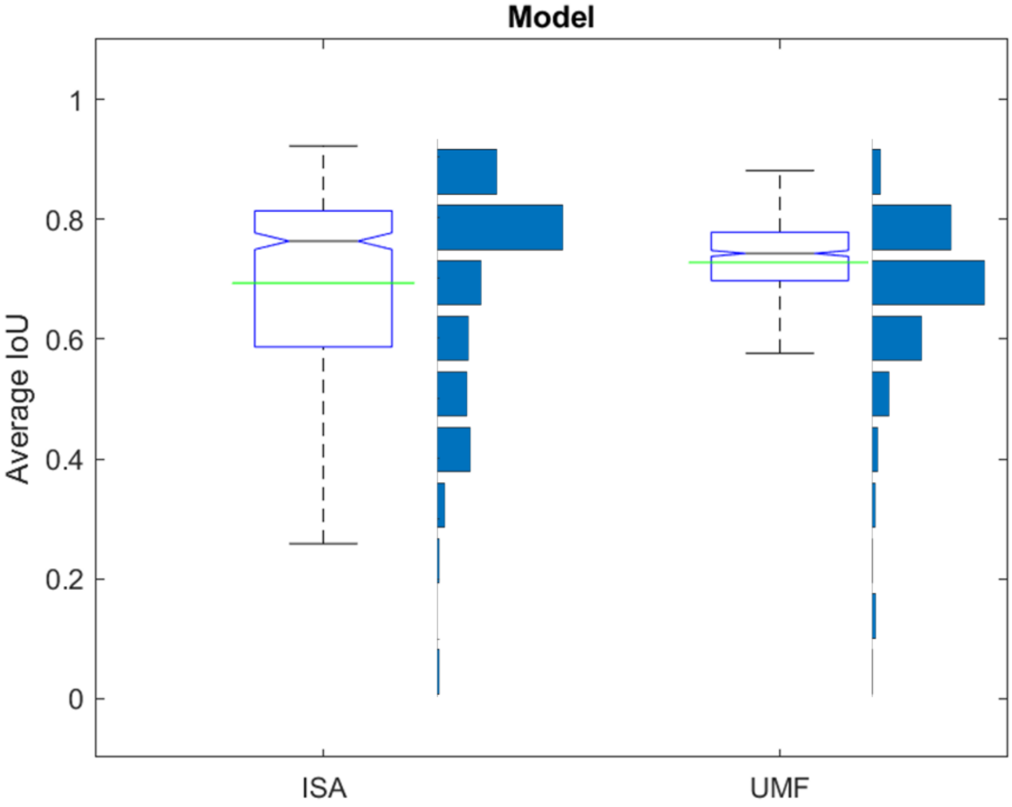
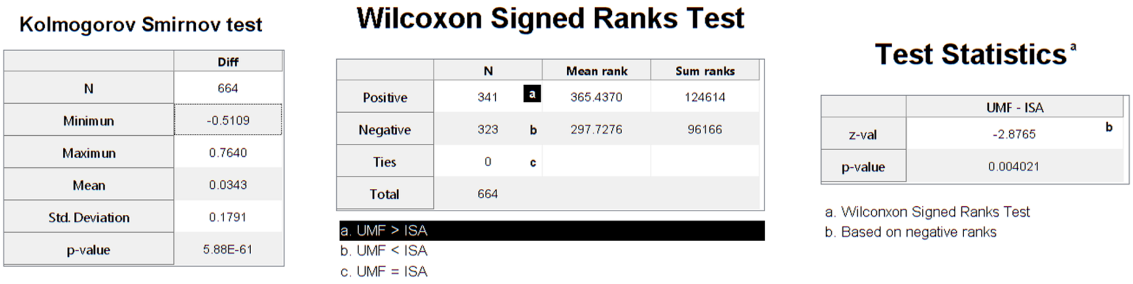


Figure 2.39: **Comparison of previous work [41] with UMF U-Net [63].** The green line indicates the mean in both graphs and the gray line indicates the median. ISA N = 664, mean = 0.6936, median = 0.7635, standard deviation = 0.1649, variance = 0.0272. UMF U-Net N = 664, mean = 0.7279, median = 0.7430, standard deviation = 0.0871, variance = 0.0076.



Kolmogorov Smirnov test

	Diff
N	664
Minimun	-0.5109
Maximun	0.7640
Mean	0.0343
Std. Deviation	0.1791
p-value	5.88E-61

Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	341	a	365.4370	124614
Negative	323	b	297.7276	96166
Ties	0	c		
Total	664			

a. UMF > ISA
b. UMF < ISA
c. UMF = ISA

Test Statistics[a]

	UMF - ISA
z-val	-2.8765 [b]
p-value	0.004021

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

(a) (b) (c)

Figure 2.40: **Statistical analyses. (a)** Normality test on the difference of methods (ISA – UMF U-Net). The p-value obtained was 5.88E-61 less than the significance value of 0.05, thus the null hypothesis was rejected and the alternative hypothesis H1 was accepted (the data did not come from a normal distribution). Once the alternative hypothesis was accepted, the Wilcoxon signed-rank test was used to evaluate both methods. **(b)** The Wilcoxon signed-rank test table shows the difference between two related samples across positive, negative and tied ranks. **(c)** The p-value obtained with the Wilcoxon rank test was 0.0040 less than the significance value of 0.05, thus concluding there was a statistically significant difference between both models.

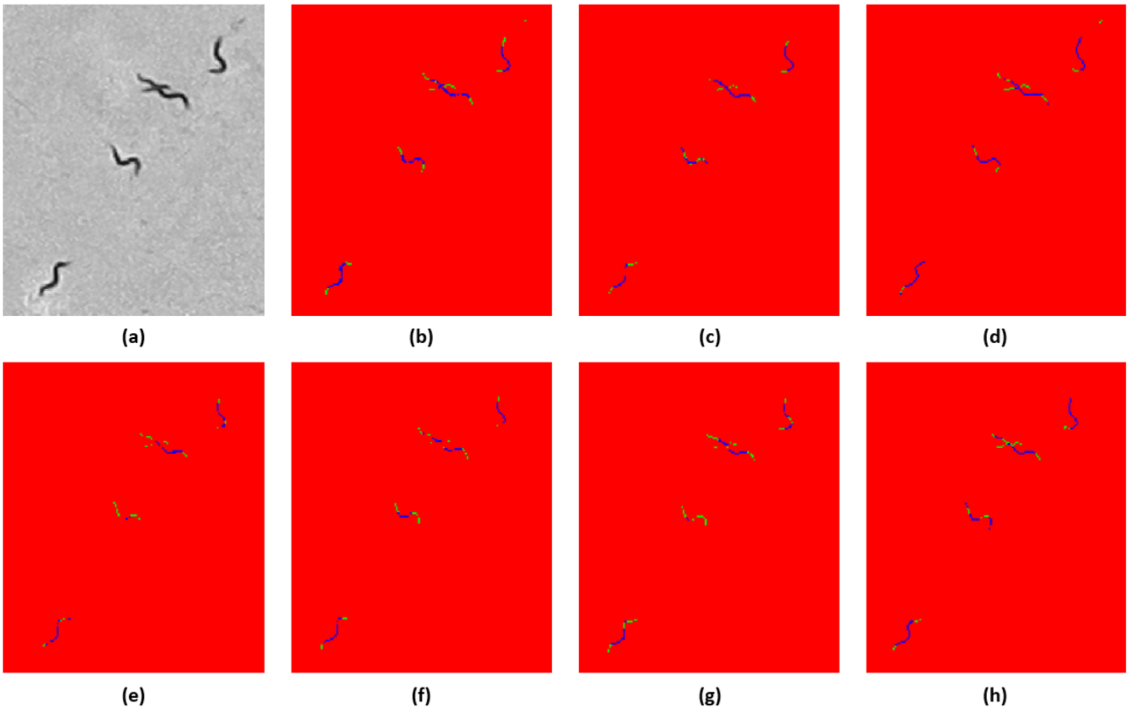
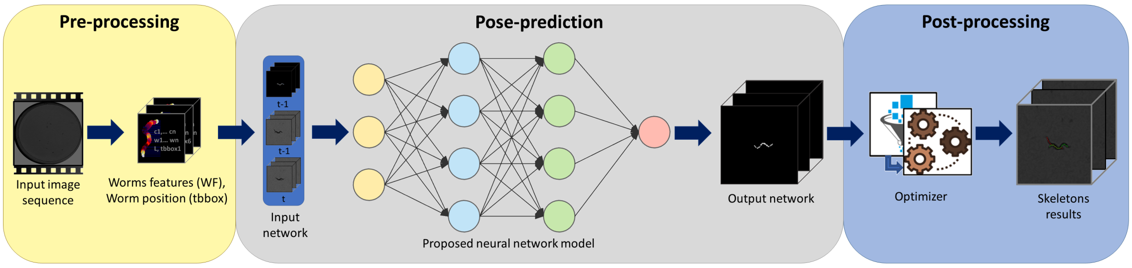


Figure 2.41: **Comparison of skeletons obtained with the different U-Net architectures.** An image was selected and results were obtained coded for all the different architectures, then the same section was cropped in all the images. **(a)** Grayscale image. **(b)** Result with U-Net standard [79], **(c)** Result with Alexandre's U-Net [1], **(d)** Result with UMF U-Net [63], **(e, f, g, h)** Result with SmaAt-UNet [93] (SmaAT Ds, SmaATDs At, SmaATDs At 4CBAMs and SmaAT, respectively).

2.5 Paper5

Automatic segmentation of *Caenorhabditis elegans* skeletons in worm aggregations using improved U-Net in low-resolution image sequences.

Pablo E. Layana Castro[1], **Antonio García Garví** [1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain



Graphical abstract.

Highlights

- A *C. elegans* skeleton prediction method is proposed based on U-Net architecture for multi-worm aggregation with low-resolution images.
- A new method for generating a custom-labeled dataset for different *C. elegans* aggregation with low-resolution images is proposed.
- A neural network has been trained with a low-resolution synthetic image and successfully tested in the domain of real images.
- A *C. elegans* tracking method is proposed based on U-Net architecture and optimizer with filter possible solutions for multi-worm aggregation with low-resolution images.

Automatic segmentation of *Caenorhabditis elegans* skeletons in worm aggregations using improved U-Net in low-resolution image sequences.

Pablo E. Layana Castro[1], **Antonio García Garví** [1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Layana Castro, P. E., Garví, A. G., Sánchez-Salmerón, A. J. (2023). Automatic segmentation of *Caenorhabditis elegans* skeletons in worm aggregations using improved U-Net in low-resolution image sequences. Heliyon. https://doi.org/10.1016/j.heliyon.2023.e14715.

2.5.1 Abstract

Pose estimation of *C. elegans* in image sequences is challenging and even more difficult in low-resolution images. Problems range from occlusions, loss of worm identity, and overlaps to aggregations that are too complex or difficult to resolve, even for the human eye. Neural networks, on the other hand, have shown good results in both low-resolution and high-resolution images. However, training in a neural network model requires a very large and balanced dataset, which is sometimes impossible or too expensive to obtain. In this article, a novel method for predicting *C. elegans* poses in cases of multi-worm aggregation and aggregation with noise is proposed. To solve this problem we use an improved U-Net model capable of obtaining images of the next aggregated worm posture. This neural network model was trained/validated using a custom-generated dataset with a synthetic image simulator. Subsequently, tested with a dataset of real images. The results obtained were greater than 75% in precision and 0.65 with Intersection over Union (IoU) values.

keyword: *Caenorhabditis elegans*, skeletonizing, synthetic dataset, low-resolution image, U-Net.

2.5.2 Introduction

Caenorhabditis elegans (*C. elegans*) are widely studied animal models, their behavior provides valuable information for many researchers in the field of biology [5]. This animal model is attractive to study and evaluate the different effects of its mutations such as the treatment of aging, age-related pathologies and neurodegenerative disorders in at advanced ages [58].

Currently, there are many semi-automatic or automatic applications for predicting worm pose, and tracking single or multiple worms. These applications that use traditional computer vision techniques have shown good results in pose prediction [90, 41], behavior analysis [6, 83, 73], locomotion [8], location of multiple worms [17], calculation of the average speed of multiple worms [70], and even using low-cost tools [45]. Pose prediction is a very important technique in tracking applications because it allows us to automate all these types of tests mentioned, lifespan and healthspan, among others. However, it is often a challenge when there is overlap and contact between worm bodies or plate noise. Some applications often have trouble resolving these cases [100, 33, 38]. In a previous work [42], we showed that using an optimizer with temporal information of the images, a better solution could be obtained in these cases of aggregation.

To train a neural network, a large dataset is necessary, and in certain applications such as *C. elegans* as well as some animal models or living organisms, it is usually difficult to obtain. Different methods have been proposed to try to address this issue, for instance some applications propose data augmentation techniques using synthetic images, while others propose a purely synthetic dataset. Both methods have obtained satisfactory results on high-resolution images. Currently, there are software applications that use neural networks for worm pose prediction. For multiple worm cases, [98] recognizes live/dead worms, while [8, 51] combine neural networks with traditional machine vision techniques to separate connected worms. These mentioned cases are performed on static images and images of high-resolution.

In this work, a novel method is proposed for predicting *C. elegans* poses (skeletons) for multi-worm aggregations in low-resolution gray image sequences. Although working with low-resolution images of complete 55mm petri dishes allows automating many assays for motility, lifespan, etc, it also presents a limitation in extreme cases of aggregation. For example, in cases of aggregations with a large amount of noise, where a large part of the worm's body has been lost, the proposed method obtained a greater loss of identity than in the other cases. Certain cases of parallel aggregation, where the worms are closely linked, are also challenging, as the identity of one of them can be lost. The proposed method uses deep neural networks and traditional image processing techniques to predict worm skeletons and track them throughout a sequence of images. The neural network model was trained and validated with a purely synthetic low-resolution gray image dataset and tested with a real image dataset. Our synthetic images aim to simulate interactions between worms, as well as individual movements, while wormpose [25] only simulates individual worm behaviors. Specifically, the generated behaviors are parallel aggregations and cross aggregations. Simulating these behaviors is very interesting since unbalanced many of these cases are not available in the real dataset and they are the most difficult to solve in detection and monitoring applications. Although our synthetic images generates aggregation behaviors between worms, this method could only be applied to low-resolution images.

2.5.3 Methods

Caenorhabditis elegans strain and culture conditions

Two types of adult-young *C. elegans* strains were used: N2 and *CB1370 daf-2(e1370)*. Both strains were obtained from worm eggs incubated at 20ºC in NGM plates measuring 55 mm in diameter. To prevent reproduction, FUDR (0.2 mM) was used. *Escherichia coli* strain OP50 was used as a standard diet. To avoid fungal contamination, fungizone (1µg/mL) was added and the plates were closed with a lid. The standard method [88] was followed to remove contaminated plates. Plates with 10, 15, 30, 60, and 90 nematodes were cultured to obtain greater variability and analyze different types of behavior (aggregation of two or more *C. elegans*, aggregations with plate noise, and occlusions).

Image capture system

First, a laboratory operator collected the plates from the incubator and inserted them into the capture system. The program captured the 30-image sequence at an interval of one second per image. The ambient temperature inside and outside the incubator was 20°C to avoid condensation. All experiments were performed with lids and *E. coli* OP50 seeded in the center of the plates to prevent worms from moving out of the field of view by climbing on or near the plate edges. The worms are adult-young and a time period of one second per image, or 1Hz, was enough to capture their movements. In our images, the length of each worm varies from 20 to 40 pixels. At this stage of their life their movement is slower than when they are younger. This movement is between 1 to 15 pixels in our image settings.

The capture system used a Raspberry Pi v1.3 RGB camera (Rasberry Pi Foundation), OmniVision OV5647 with a resolution of 2592 × 1944 pixels, pixel size of 1.4 × 1.4 µm, field of view of 53.50° × 41.41°, optical size of 1/4″ and focal ratio of 2.9. The lighting system was based on a 7″ Raspberry Pi screen with a resolution of 800 × 480 at 60 fps, 24-bit RGB color, and a Raspberry Pi 3 as a processing unit. The lighting system used active backlighting, a technique that reduces the variability of the captured images, keeping the grayscale more constant. This technique has proven to be effective in *C. elegans* image capture systems [67, 68]. SiViS [67] is a *C. elegans* monitoring machine that can analyze *C. elegans* cultures using standard Petri dishes seeded with *E. coli*. SIVIS [67] has a flexible and compact platform design, which allows it to be adapted to longevity assays, as well as other studies of *C. elegans*, such as toxicity, motility, and behavior. The SIVIS [67] image capture system (Figure 2.43) is open hardware and software, whose assembly and image capture processes are described in detail in [67].

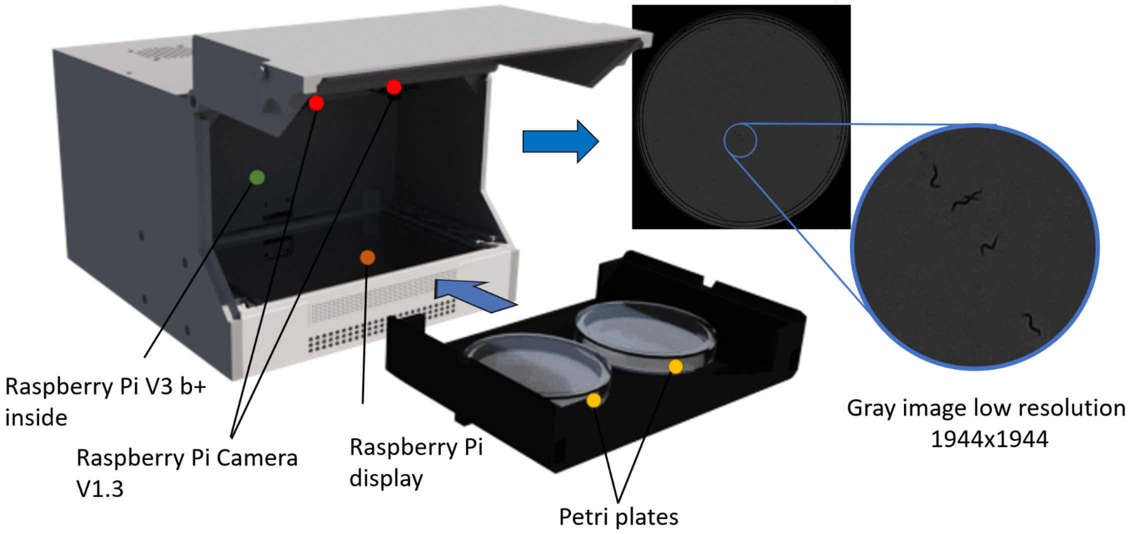


Figure 2.43: **SIVIS image capture system.** The figure presents all the parts and elements of the SIVIS [67] capture system, such as an example of a grayscale image captured by the system.

Synthetic image acquisition method

To train and validate the neural network we created a dataset of synthetic images based on 600 sequences of 30 images with worm aggregations. The synthetic images were generated in two stages. The first stage, Simulator, was used offline before training to generate *.PTS files. PTS files are text files and contain information on color, width, and location (X, Y) of each pixel of the worm's skeleton. The second stage was used during the training/validation phase and used the information from these *.PTS files to create low-resolution grayscale images of worms.

To create the *.PTS files (worm aggregate sequences) the Simulator used images of empty Petri dishes (free of worms), and individual moving worm sequence files. Each worm file (XML file) contained skeleton point information ordered during a sequence (30 skeletons), it also contained color information and width values for each skeleton point. The simulator randomly selected worm files and produced aggregations by randomly matching 1 skeleton point from each worm. Finally, an X-Y position and rotation angle were also randomly generated to position the worms within the Petri dish image.

In total, 628 worm files were used to produce all the new 30-move sequences. These worm files were manually tagged, by a tagger person, from the sequence of movements of worms moving freely in Petri dishes.

Proposed neural network method

This work aims to obtain the current pose of a worm within an aggregation using a sequence of two gray images (previous and current image) and a binary image of the skeleton of the worm of interest (previous skeleton image). The proposed neural network model (Figure 2.44) consists of six neural network blocks, three encoder blocks marked in Figure 2.44 by blue, orange, and dark orange colors correspondingly, one Long Short-Term Memory block (LSTM) marked in the Figure 2.44 by green, one Linear block marked in the Figure 2.44 by violet, and one decoder block marked in the Figure 2.44 by yellow. The outputs of the encoders generate feature blocks for each input image. The feature blocks in the output of the gray image encoders are exploited by the LSTM to predict future pose. The linear block matches the feature depth of the LSTM output with the binary image encoder output feature block in order to concatenate features of the individual of interest. Finally, the decoder block reconstructs the output image from the input features and the concatenation of features during encoder compression.

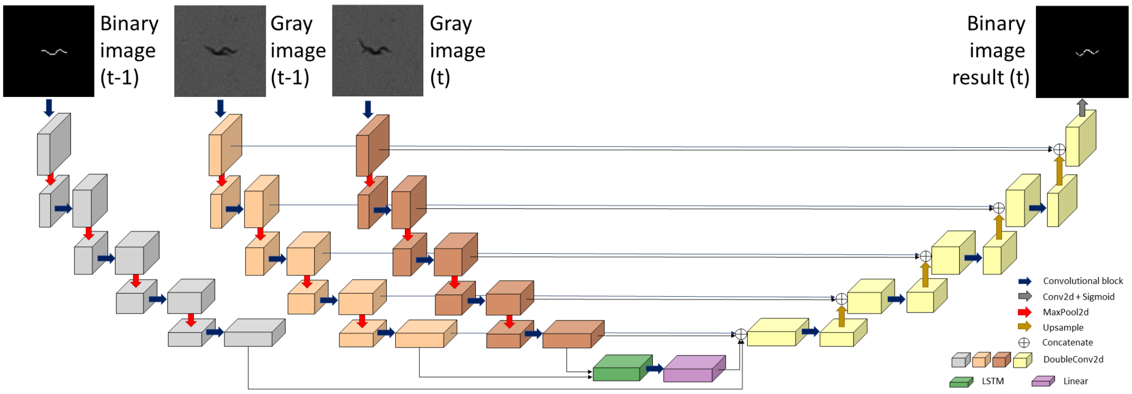


Figure 2.44: **Proposed neural network model.** Blue, orange, dark orange, and yellow blocks correspond to the encoding and decoding blocks of Alexandre's U-Net convolutional neural network model [1]. The green block corresponds to the LSTM block and, lastly, the violet block corresponds to a linear block.

Alexandre's U-Net convolutional neural network architecture [1] was used for the encoder and decoder blocks, this architecture is similar to the traditional U-Net [79] but after each convolution2d layer a BatchNorm2d layer is added. Convolution2d (torch.nn.conv2d) and BatchNorm2d (torch.nn.BatchNorm2d) are functions of Pytorch libraries. Convolution2d applies filter convolutions on its 2D input, while BatchNorm2d normalizes the input data for each channel independently. For all the 2d Convolution layers in the encoders as well as in the decoder, kernel_size=3 and padding=1 were used, except for the last 2d Convolution layer, where kernel_size=1 and padding=0. As mentioned in [29] applying a batch normalization step after each convolutional layer allows regularization of the data, such as better convergence, thereby reducing the internal covariate change in the dataset.

The input of all the encoder blocks as well as the output of the decoder block were 100x100 pixels images. This ensured that the next worm move (network output), measuring 40 pixels long, would fall within the window.

Training/validation and testing method

To evaluate the proposed model, two datasets were used, a dataset of synthetic images for the training and validation stages, and a dataset of real images to test the results in cases of multi-worm aggregation, and aggregation with noise (Figure 2.45).

The image was divided into fixed windows of 100x100 pixels due to the input image dimensions (1944x1944 pixels) and the limitations of the hardware to train, validate and test the proposed network model (78.76M parameters). Each window contained a worm in the image center.

A 3x3 normalized box filter was applied using the OpenCV tool (cv.blur(3,3)) for all real and synthetic grayscale image windows. This filter uses a kernel (3x3 matrix) to average the pixels that lie in that area and replace the central element with the calculated average. This filter was essential to bring both datasets (synthetic and real) closer to the synthetic-real image domain. It allowed to generalize both domains. The 3x3 size is the minimum size within the OpenCV tool and allowed to perform this operation.

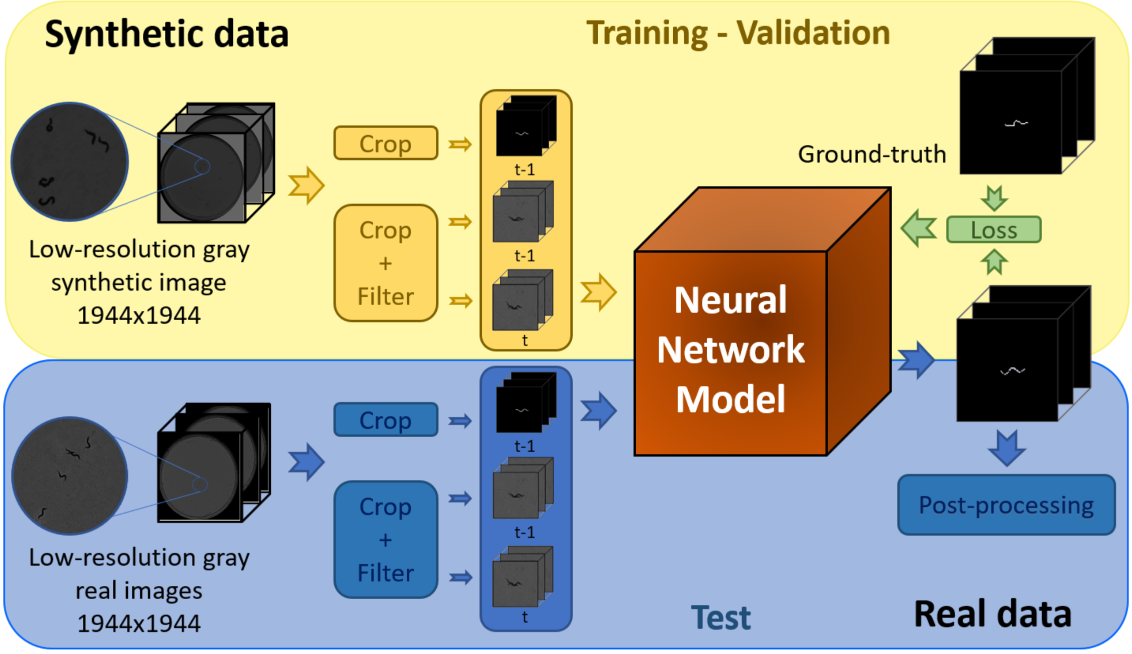


Figure 2.45: **Synthetic and real dataset pipeline.** A dataset of synthetic images was used to train and validate the network. The trained network was used to test the real image domain dataset.

The loss function to train/validate the network was the PyTorch function “BCELoss()”. Binary cross entropy loss or BCELoss() is a Pytorch loss function for binary classification i.e. for single unit output classification. The metrics to evaluate the real dataset are detailed in "Results validation methods".

Post-processing method

This method (Figure 2.46) comprised three stages, the first stage of possible skeleton solutions was generated using the method proposed in [41]. This method generated possible skeleton solutions using endpoints and intersection points in the improved skeleton image. The improved skeleton image was created using information of the worm widths, to do so the algorithm set pixels in the background (pixels equal to zero) when the thickness in a segmentation exceeded the maximum width of the worm. The maximum width of the worm is the distance value from the widest edge of the worm's body to its opposite. This value is usually found in the center of the worm's body. In our images, the length of each worm varies from 20 to 40 pixels, while the width varies from two to four pixels. This enabled skeletons to be separated during an aggregation or when self-occluding.

The post-processing second stage used the skeleton predictions obtained from the output of the neural network model and the possible skeleton solutions of each worm to filter those solutions with a coincidence percentage below 65% (F=0.65). The F value selected was the value closest to 1 on a precision-recall curve.

The third and final stage used the optimizer proposed in [42] to obtain skeletons for each worm in an aggregation. The optimizer algorithm uses three criteria to evaluate all possible combinations of each worm skeleton in an aggregation. The worm skeletons used are obtained from the filter of possible solutions. The criteria are described in more detail in [42] and are: overlap with the previous pose, completeness of the segmentation, and color. The best possible solution is the minimum value of the sum of these three criteria in all possible combinations.

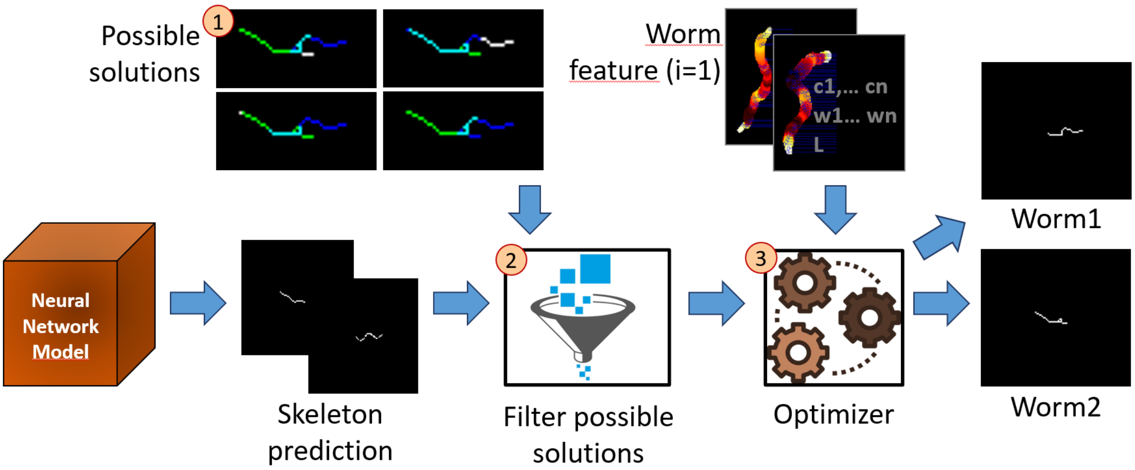


Figure 2.46: **Pipeline of the post-processing method.** The post-processing method generates possible skeleton solutions for all worms in the actual skeleton image. Likewise, an optimizer proposed in [42] uses the filtered solutions, employing the predictions of the proposed neural network model and the features of each worm obtained in the first image to obtain the most probable skeletons for each worm.

Proposed tracking method

The tracking method consisted of going through an image sequence and predicting the current pose(skeleton) of each worm using the proposed neural network and a post-processing method to enhance skeletons. The input to the neural network included three cropped images measuring 100x100 pixels. Two images corresponded to the previous state (grayscale image and binary image of the worm skeleton) and one current grayscale image. The tracking method started at the second image because the first image was used to extract Worm Features (WFs), as well as being used to save the previous grayscale images (Img1) and skeleton images (SKL), and bounding boxes (tbbox). After each post-processing step an update process used the results obtained from skeletons and bounding boxes to move all worm windows (tbbox) to their new positions and to update the previous images. The process was repeated until all the images were finished, as well as shown in Algorithm 1.

Algorithm 1 ***Tracking algorithm.*** *The algorithm shows the sequence taken by the images to track each worm. First, the features of each worm (WF | x |), the previous gray cropped images (Img1 | x |), and their skeleton images (SKL | x |) are obtained. This information is used in the next stage (i>1), where the current cropped images (Img2 | x |) are acquired and the poses of each worm are predicted (Pred | x |). A post_processing function generates all possible skeletons for each worm and filters them using the predictions to find the best solution. Information from tbbox, SKL and Img1 is updated with the obtained results. The procedure is repeated until the image sequence is complete.*

```
Input: total_images, total_worms, path_network, gray_images
Net= Load(path_network)
  for $i = 1$ to total_images do
      img = Get gray_images(i)
      for $x = 1$ to total_worms do
          if $i == 1$ then
              tbbox[x] = Get bounding box                ▷ bounding box for each worm
              Img1[x] = crop(img, tbbox[x])
              SKL[x] = Get skeleton worm          ▷ Binary image of the worm skeleton
              WFs[x] = Get worm's feature  ▷ Colors of each skeleton pixel, worm length
          end if
          if $i > 1$ then
              Img2[x] = crop(img, tbbox[x])
              Pred[x] = Net(Img1[x], Img2[x], SKL[x]) ▷ Network predictions of all worm skeleton
          end if
      end for
      if $i > 1$ then
          SKL_opt, tbboxN = post_processing(img, Pred, WFs)     ▷ Binary image of all worm skeleton and updated tbbox
          Update:                                                  ▷ Update variables
              tbbox = tbboxN
              SKL = SKL_opt
              Img1 = crop(img, tbbox)
      end if
  end for
```

In the first image, after obtaining the worm skeletons, the bounding boxes were obtained for each worm, placing the skeleton of each worm in the center of a 100x100 pixel window. The bounding boxes of each worm were used to crop 100x100 pixel windows for each worm and for each skeleton. Bounding boxes, as well as gray images and skeleton images, were saved in lists (tbbox, Img1, SKL, respectively). Before saving the cropped gray images, a 3x3 normalized squares filter was applied. To obtain the features of each worm, the cropped binary image of skeletons (location of pixels) was used and the color and width of each pixel of the skeleton was extracted from the cropped gray image (Img1). All this information on length, color and width was saved in a list for each worm (WFs).

Results validation method

The Intersection over Union (IoU) index was used to evaluate the precision of results obtained with the real dataset [39]. This metric, as its name indicates, measures precision by dividing the intersection of ground-truth areas and the results obtained for the union of these areas (Equation 2.30).

$$IoU = \frac{\sum P_{w1} \bigcap P_{w2}}{\sum P_{w1} \bigcup P_{w2}} \tag{2.30}$$

The ground-truth (manual labels) and the results obtained correspond to worm skeletons dilated with a disc of 2 to recover the shape of each worm (Figure 2.47a and Figure 2.47b respectively). The skeletons are sequences of pixels arranged on the centerline of each *C. elegans*. To obtain the skeletons of the manual labels (ground-truth), an annotator manually selected each point of the skeleton pixel by pixel, while the skeletons corresponding to the results were obtained after post-processing. For the examples in Figure 2.47C, d, e the IoU results were 0.7004, 0.8204 and 0.9409, respectively.

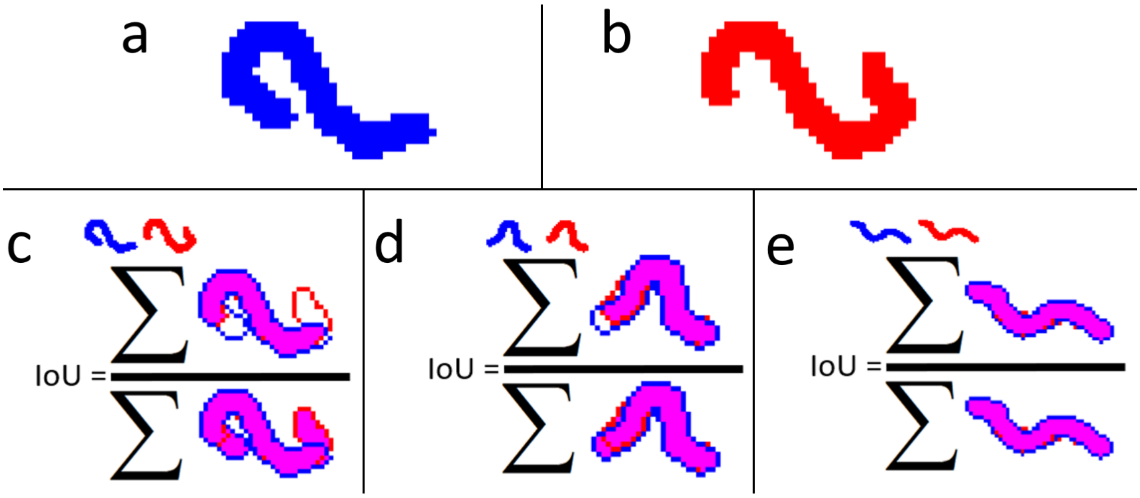


Figure 2.47: **IoU index.** This index evaluated how accurate the response of the method was to the reference (ground-truth). The higher the IoU value, the more accurate the response to the reference. Predicted skeletons are not always exact, usually one or more pixels are offset from the reference (ground-truth). These offsets usually occur when the width of the worm is even, while when the width is odd, the skeleton pixel is the central one. To obtain a better measurement of the results, the evaluation was carried out on the dilated worm skeletons with a 2-pixel disc to recover the shape of each worm. (**a**) Body reconstructed from the manually labeled skeleton. (**b**) Body reconstructed from the skeleton obtained from the optimizer output. (**c-e**) Examples of validation of reconstructed skeletons, at the top of each example you can see the reference (blue), ground-truth, as well as the response of the automatic method (red).

To obtain the value F=0.65 (percentage of coincidence) used in the filter of possible solutions, two metrics were used: precision and recall (Equation2.31, Equation2.32 respectively). These metrics evaluated the performance of all the results obtained with the manual labels of the real dataset (ground-truth). These metrics, as well as the IoU index, were evaluated using the dilated skeletons of the manual labels and the predictions (Figure 2.48c and Figure 2.48d respectively). To obtain each of these metrics, four parameters were used: *TP* (true positives), *FP* (false positives), *TN* (true negative), *FN* (false negative) Figure 2.48b. *TP* was obtained when ground-truth and prediction equaled one. *FP* was obtained when ground-truth equaled zero and prediction equaled one. *FN* was obtained when ground-truth equaled one and prediction equaled zero. Lastly, *TN* was obtained when ground-truth and prediction were equal to zero. The prediction and recall results for the examples in Figure 2.48a, g were: precision=0.8058, recall=0.8426 (Figure 2.48e, f) and precision= 0.9384, recall=0.8671 for (Figure 2.48h, i).

$$Precision = \frac{TP}{TP + FP} \tag{2.31}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.32}$$

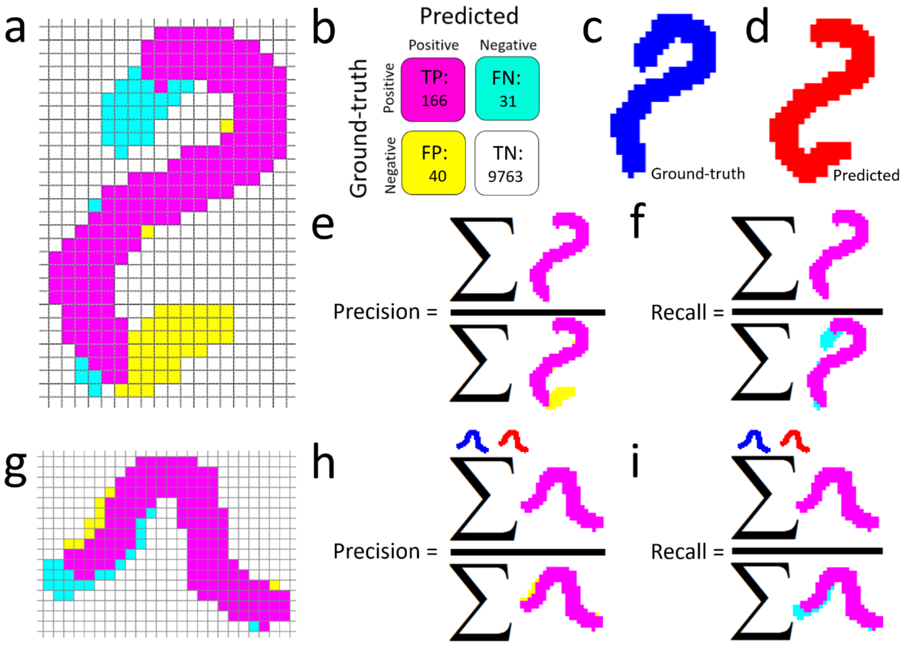


Figure 2.48: **Precision-Recall metrics.** These metrics evaluated the results obtained with respect to the reference (ground-truth) to select the *FP* value used by the filter to eliminate possible solutions. (**a, g**) The *TP*, *FP*, *FN* and *TN* parameters used to calculate the precision and recall metrics, are shown in color (magenta, yellow, cyan and white, respectively). (**b**) *TP*, *FP*, *FN* and *TN* values for the example in Figure 2.48a. (**c**) Reconstructed body of the manually labeled skeleton for the example in Figure 2.48a. (**d**) Reconstructed body of the skeleton obtained from the optimizer response for the example in Figure 2.48a. (**e, f, h, i**) Examples of the evaluation of precision and recall metrics of the reconstructed skeletons for the examples in Figure 2.48a, g.

The Matlab 2018b Machine Learning Toolbox software was used to compare the statistics between the proposed tracking method with a filter equal to zero and a filter equal to 0.65. The Kolmogorov-Smirnov Test (for large samples greater than 50 data) was used to check data normality while the Wilcoxon Signed Rank Test was used to evaluate the statistical significance of the results with the F=0.65 filter.

2.5.4 Experiments and results

A synthetic dataset was used to train/validate the proposed neural network model and a dataset of real images to test all the results. For the synthetic dataset, 600 sequences of 30 images (18000 images) were simulated, 70% of the sequences were used for training, that is, 420 sequences (12600 images), and the remaining 30%, 180 sequences (5400 images), for validation. Each simulated sequence image had two worms per plate, simulating different aggregation events. The images of each sequence were taken two by two starting from the first, that is, image1 with image2, image2 with image3, and so on until the end.

The hardware used for training and validation was a Gigabyte Technology Z390 AORUS PRO machine, Intel(R) Core (TM) i9-9900KF CPU @ 3.60GHz x16 with 32GB of RAM and NVIDIA GeForce RTX 2080 Ti graphics card with 4352 Cuda cores, Ubuntu 19.04 64-bit operating system. The implementation was performed in a Python version 3.7.5 environment, using the Pytorch 1.18, OpenCV 4.5.4 and SWIG 3.2 libraries. The training and validation of the proposed neural network model took about 36 hours with the aforementioned hardware and software. The hyper-parameters used for the training and validation phase were Batch_size = 10, num_workers = 10, maximum epoch = 100. The optimizer used was ADAM with a learning rate = 0.0001, betas = [0.95, 0.999], eps = 1e-8, BCELoss() as loss function for the training/validation step, and finally, the scheduler used was the ReduceLROnPlateau scheduler with hyperparameter mode = 'min and patience = 2.

To save the network trained during each epoch, the criterion was chosen if the average validation value was greater than the previous one, it was saved over the previous one, overwriting the saved network. The best average results were 0.001017 and 0.001935 for training and validation, respectively. The results obtained in each of the training and validation epochs are shown in Figure 2.49.

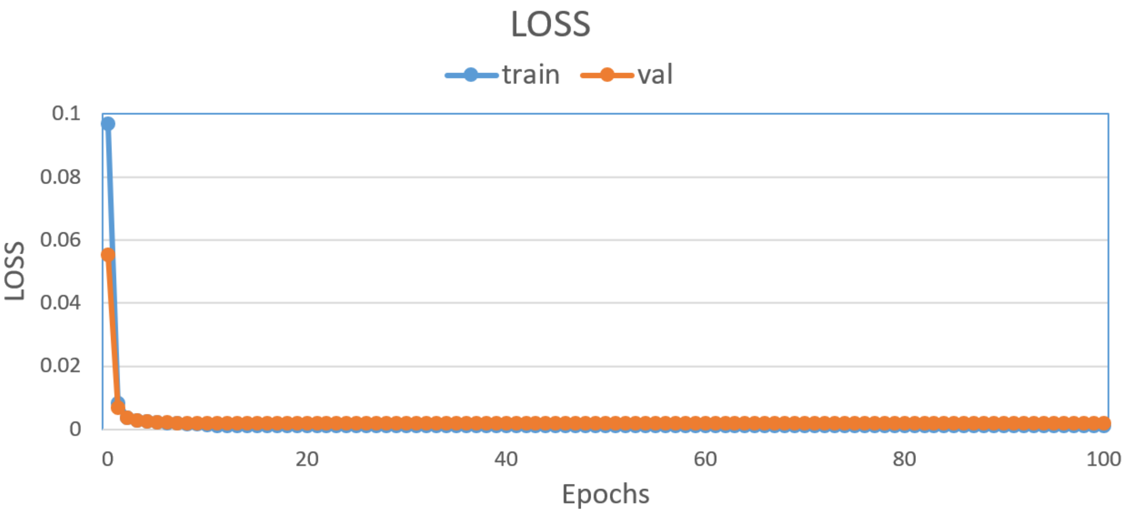


Figure 2.49: **Training/validation graph.** This figure shows the average results (y-axis) of 5400 images for each epoch (x-axis). The training (blue line) and validation (orange line) results were obtained using the PyTorch BCELoss() function.

For the real image dataset, 4500 Petri dish images (150 sequences of 30 images) were analyzed with subsequent selection of those difficult cases where worms aggregated with each other or aggregated with dish noise. To obtain greater variability of aggregations, image sequences with 10, 15, 30, 60 and 90 worms per plate were used. As mentioned in

[89, 66] the higher the number of worms, the higher the probability of aggregation events per plate. The total number of poses found was 788, 325 for cases of multi-worm aggregation and 463 aggregations with plate noise. We defined as plate noise all segmentation errors due to opaque debris and dark objects on the plate. Usually, these errors are dark with worm-like shapes, their grayscale is almost constant, so they are distinguished from worm bodies, which have different characteristics.

To obtain the best filter value (F), its value was varied from 0.00 to 1.00 in steps of 0.01. For each value, average precision and recall values were obtained for each skeleton for each prediction in multi-worm aggregations and aggregations with noise (788 aggregations). The selected filter value was the one whose precision and recall values were closest to 1. Table 2.13 shows the average precision and recall values obtained for a filter equal to 0 (predictions made without a filter) and with a filter equal to 0.65 (best precision and recall results values), the average precision and recall values for the cases of multi-worm aggregation and with plate noise are also shown.

Table 2.13: **Comparison of the average values of precision and recall metrics using a 0.65 filter and 0 filter.** A total of 788 worm poses were used, of which 325 poses were multi-worm aggregations and 463 noise aggregation poses. The best results were obtained using the 0.65 filter.

		Avg. Precision		Avg. Recall	
Cases	Total pose	F=0.00	F=0.65	F=0.00	F=0.65
Aggregation	325	0.7348	**0.7893**	0.7286	**0.7832**
Noise	463	**0.7892**	0.7877	**0.7573**	0.7562
Total	788	0.7649	**0.7884**	0.7445	**0.7683**

To compare the proposed next-pose prediction method (next skeletons) with previous works [42], the filter of possible solutions to zero was used, which enabled comparison under the same conditions. The metric used to measure all the results was the IoU index, as in previous works. The average values without filter (filter equal to zero) are similar to the average values of previous works [42]. Table 2.14 shows the comparison of both methods (with filter and without filter), as well as all possible solutions and the filtered solutions with the best filter value obtained for all analyzed aggregation cases. As can be seen in the last column, since the filter reduces more than 75% of possible solutions Table 2.15, the optimizer only had a load of less than 25% (difference between possible solutions and filtered solutions). The results using the best filter value show an improvement over previous work [42], such as a considerable reduction in computational cost.

Table 2.14: **Comparison of the average values using the IoU Index for 0.65 filter and 0 filter.** This table shows the average values obtained using the IoU index with filter equal to 0.65 and without filter (IoU values similar to previous works [42]). The cases evaluated with the filter values were multi-worm aggregations (325) and aggregations with noise (463).

		Avg. IoU		Standard Deviation	
Cases	Total pose	F=0.00	F=0.65	F=0.00	F=0.65
Aggregation	325	0.6033	**0.6561**	0.2106	**0.1503**
Noise	463	**0.6507**	0.6504	**0.1856**	0.1876
Total	788	0.6295	**0.6529**	0.1985	**0.1718**

Table 2.15: **The filtered solutions column shows the total possible solutions filtered using the coincidence filter.** The difference between the possible solutions column and the filtered solutions column gives the total number of possible skeletons processed by the optimizer.

Cases	Total	Possible solutions	Filtered solutions
Aggregation	325	2505	1929
Noise	463	2552	1964
Total	788	5057	3893

Statistical analysis was performed to compare the results obtained without the filter (previous work [42]) and using the filter of possible solutions (proposed work). The p-value (0.05) was used to evaluate statistical significance. Normality was first analyzed with the Kolmogorov-Smirnov Test to assess the difference between both methods. This test is used for large sample sizes (n>=50) and if the p-value was greater than or equal to the 5% level of significance (0.05), the null hypothesis H0 was accepted (the data came from a normal distribution), otherwise, the alternative hypothesis H1 was accepted (the data did not come from a normal distribution). The results indicated that they did not come from a normal distribution, p-value = p<0.001, much lower than the significance value of 0.05 (Table 2.16); therefore, the alternative hypothesis H1 was accepted, and the Wilcoxon Signed Range Test was used (Table 2.17).

Table 2.16: **Normality test on the difference of the proposed method with filter at 0 (previous traditional methods) with filter equal to 0.65 (using the neural network).** The p-value obtained was p<0.001, less than the significance value of 0.05, thus the null hypothesis was rejected and the alternative hypothesis H1 was accepted (the data did not come from a normal distribution). Once the alternative hypothesis was accepted, the Wilcoxon Signed Rank Test was used to evaluate both methods.

Kolmogorov Smirnov Test

	Diff
N	788
Minimum	-0.7527
Maximum	0.7989
Mean	0.0234
Std. Deviation	0.1505
p-value	p<0.001

Wilconxon Signed Rank Test showed that the proposed method (F=0.65) is statistically more significant than methods reported in previous works (F=0.00), Table 2.17. The p-value obtained was 0.04, which is much lower than the significance value of 0.005. Figure 2.50 shows the average IoU values for both cases, as well as the mean (green line) and median (gray line) values using a box plot.

Finally, Figure 2.51 shows an example of the results obtained for each worm in an aggregation. The proposed neural network model is used for each image series of each worm, Binary image worm1(t-1), gray image worm1,2(t-1) and gray image worm1,2(t) as network input for worm 1; and binary image worm2(t-1), gray image worm1,2(t-1), and gray image worm1,2(t) as input to the network for worm 2. Both network outputs enter the post-processing and the current worm skeletons in the aggregation are obtained: blue-cyan for worm1 and cyan-green for worm2. Other examples can be seen in: https://github.com/playanaC/Skeleton_prediction/blob/main/Demo_videos.ipynb.

Table 2.17: **Wilcoxon Signed Rank Test.** The Wilcoxon Signed Rank Test table shows the difference that exists in two related samples (F=0.00 and F=0.65) across positive, negative, and tied ranks

Wilcoxon Signed Rank Test

	N	Mean rank	Sum ranks
Positive	321[a]	326.3551	104760
Negative	297[b]	291.2828	86511
Ties	170[c]		
Total	788		

a. F=0.65 > F=0.00

b. F=0.65 < F=0.00

c. F=0.65 = F=0.00

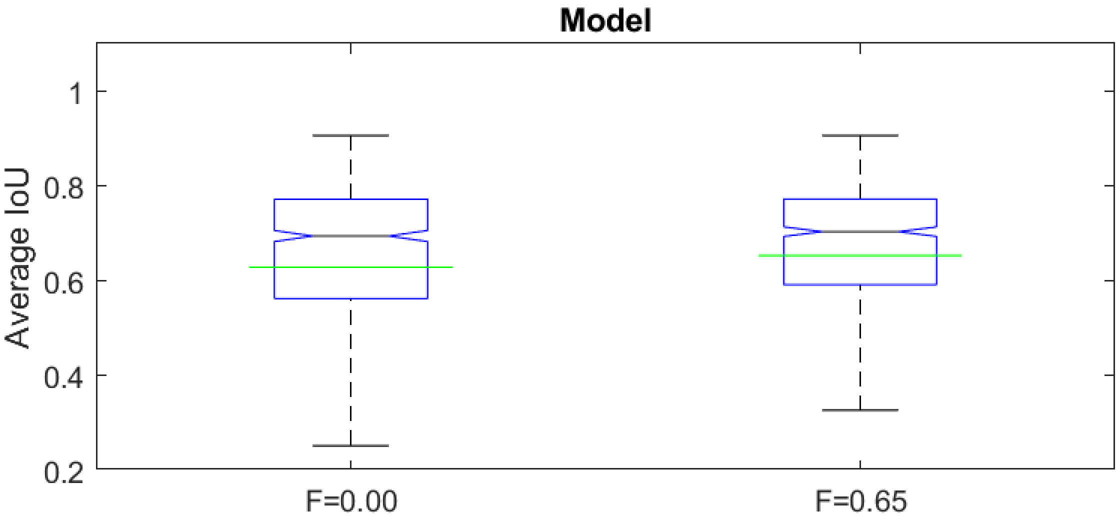


Figure 2.50: **Box plot and normality test of the difference of the 0 filter and 0.65 filter.** Box plot, the green line indicates the mean in both graphs and the gray line indicates the median. Filter at 0.00, N = 788, mean = 0.6295, median = 0.6943, standard deviation = 0.1985, variance = 0.0394. Filter at 0.65, N = 788, mean = 0.6529, median = 0.7035, standard deviation = 0.1718, variance = 0.0295.

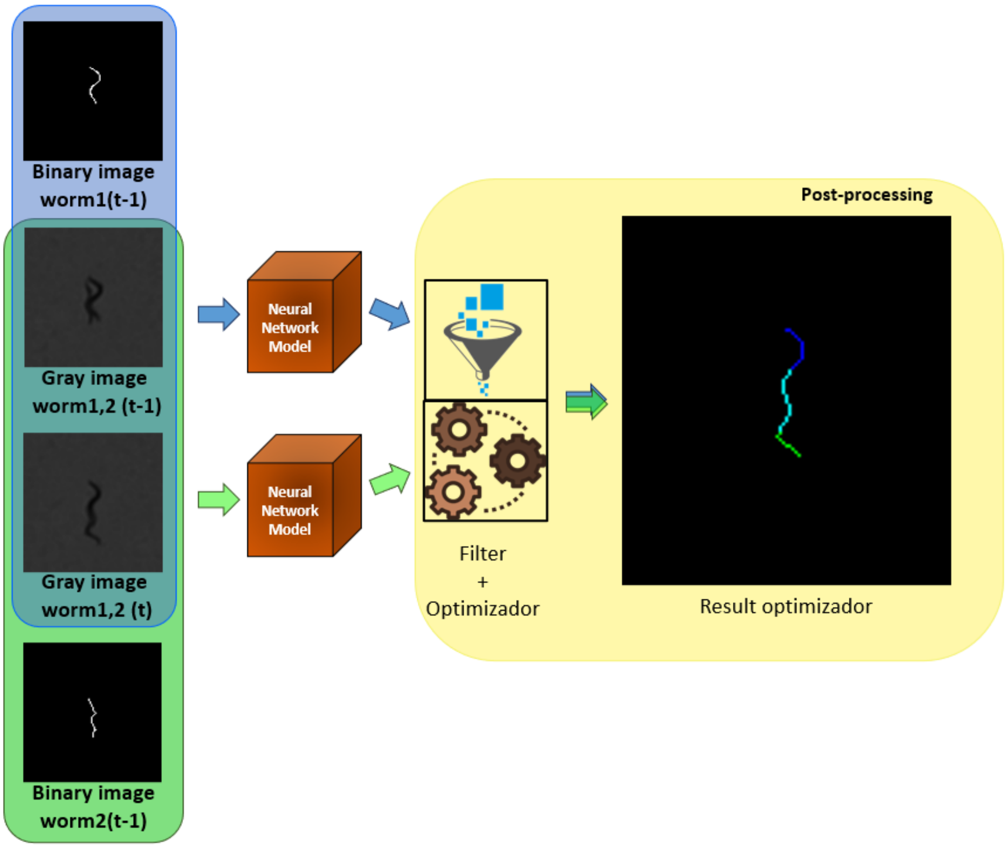


Figure 2.51: **Pipeline of a 2-worm aggregation.** The trained network is used twice to obtain the prediction for the current skeleton of each worm (worm1, worm2). The results go through post-processing and skeletons are obtained for each worm. The current skeleton of worm1 is represented by the blue-cyan pixels, while the current skeleton of worm2 by the cyan-green pixels.

2.5.5 Discussion

The time required to load the input data (images) in the network at the time of training and validating the data was considered important. Accordingly, the number of worms per plate was considered to speed up this process. Each image in the synthetic dataset image sequence had only two worms, which speeded up the normalized 3x3 crop and filtering operations during the aforementioned stages. The 3x3 size is the minimum within the OpenCV tool and allowed to generalize both domains (synthetic and real). The width of the worms in our images varies from two to four pixels, so increasing the size of this filter can excessively blur the worm and even make it disappear. Another key factor was the size of the 100x100 window, these dimensions were selected to ensure that the next movement (skeleton) of a worm, no matter how fast, is within the window.

The proposed method allows tracking multiple worms and solving aggregations of any number of worms because it predicts each of them individually, which enabled tracking 2, 3, and 4-worm aggregations from the real image dataset, see Supplementary Figures A.32, A.33, A.34, A.35. There are aggregation cases that are more complicated to solve than others, although in some instants of time there is a worm that is not perfectly matched with the ground-truth (see Supplementary Figure A.32), the other worm presents better results, and as time progresses this error is corrected leaving the worms perfectly matched. On the other hand, in cases of aggregation with noise, the same happens during the time that they are aggregated with the noise of the plate (see Supplementary Figure A.35), after this the pose prediction is more accurate. It should be noted that, although the accuracy of one of the worms is low, the identity is not lost.

As mentioned in the experiments and results sections, filter use (post-processing) reduces 77% of the possible solutions, which can translate into considerable computational savings. The total time to resolve an aggregation depends on the number of worms added, as well as potential aggregation complexity, which translates into many possible solutions. In a previous work [42], resolving these aggregations took several seconds, while in complex 4-worm aggregations, it took several minutes. The proposed method using the results of the neural network to filter possible solutions resolves 2-worm aggregations in less than 2 seconds, while it takes under 4.6 seconds to resolve 4-worm aggregations. Today there are many parallel processing or multiprocessing tools that can help speed up this processing time. Bearing this in mind and considering that the encoder and decoder can be replaced by similar models with fewer parameters, as demonstrated in [93], an online tracking application could be developed to solve cases of aggregation.

The prediction of the next postures or poses (skeleton) is really interesting in cases of aggregation since this helps to re-identify and not lose the identity of the subjects during the tracking. In this work, we have developed a technique to predict the next poses with results greater than 75% in precision and 0.60 with IoU values for multi-worm aggregations and aggregation with noise.

This technique could also be used for other types of aggregations using other small model organisms, such as bacteria cells, or zebrafish among others.

2.5.6 Conclusions

This paper presents a novel method for predicting *C. elegans* poses in low-resolution images of multiple worms in full 55mm Petri dishes. The predicted poses are images of worm skeletons and are the result of a neural network trained to predict aggregation behaviors between worms. The proposed neural network was trained and validated with custom-generated synthetic images and successfully tested on real images. A multi-worm tracking application was developed, and for this, the results obtained from the neural network were used to filter possible solutions in an optimizer algorithm proposed in previous work [42]. The results were compared with this previous work [42] and an average improvement of 2.38% in precision and 2.34% with IoU values was obtained.

On the other hand, the computational performance was also compared with previous work [42] and a greater than 75% reduction of the possible solutions generated by the optimizing algorithm was obtained, which is reflected in the computational costs. On the other hand, to determine the effectiveness of multiple worm tracking, we added the "MOTA" metric in the supplementary material (Paper5). This metric measures the performance of the tracker, i.e., both the overall accuracy and the detection of objects in the images. We compared the results with previous work [42] and obtained an average improvement of 16.96%.

Despite very good results, the prediction of the next poses presents problems in cases of high aggregation, i.e. where more than 50% of the body of one of the individuals is aggregated with another or with noise from the plate, and the prediction does not improve until it starts to separate. Future work proposes to increase the datasets, better adjust the hyper-parameters, evaluate new architectures to improve the model results, as well as post-processing methods to obtain better pose predictions.

Source code

The proposed method was developed in Ubuntu-linux 19.04 64bits using python3.7.5 with the Pytorch1.5 libraries. The C++ and OpenCv4.5 libraries were used to create the synthetic images. To use the C++ functions (synthetic dataset) from Python we employed the SWIG3.2 library. The source code is on GitHub; it is open source and can be downloaded from the repository at https://github.com/playanaC/Skeleton_prediction. In this link there are a demo of the image simulator (synthetic dataset), demo of the proposed neural network method such as demo videos. The dataset with all the aggregation experiments can be downloaded from https://active-vision.ai2.upv.es/wp-content/uploads/2021/02/dataset_skeletons.zip. The imaging system, parts, and assembly can be downloaded from https://github.com/JCPuchalt/SiViS.

Chapter 3

Discussion

This thesis used traditional image processing techniques as well as deep learning techniques. Both techniques had their advantages and disadvantages in the postures-tracking of *C. elegans*. On the one hand, traditional computer vision techniques were transparent, i.e. all the functionality of the code was explainable, being able to adjust its parameters to adapt to changing scenarios. However, this technique required many algorithms/functions, making image processing slower. Papers 1 (section 2.1) and 2 (section 2.2) used these techniques and although they obtained values of 0.7 with the IoU metric, additional image processing was required to filter out noise (segmentations similar to the *C. elegans*) as well as to detect and track all the *C. elegans* in image sequences. Deep learning techniques, on the other hand, allowed faster results in *C. elegans* detection (section 2.3), as well as pose prediction (section 2.5). However, large datasets and as much data variability as possible were needed to train these models. Paper 3 (section 2.3) used a large dataset and increased rotation, mirror view, and color change data to increase that variability, while paper 5 (section 2.5) proposed a synthetic image generator to train a deep learning model. The responses of these deep learning models have a hit probability that depends on the training dataset, any change in this dataset implies a new training.

Posture-tracking of *C. elegans* is not an easy task to solve, either using traditional techniques or deep learning techniques. The results depend on the complexity of the postures, especially during aggregations or coiling, where postures are usually difficult to recognize even for humans. *C. elegans* are model animals with very flexible bodies, capable of infinite postures, which complicates their posture-tracking. To solve this problem, this thesis has proposed different skeletonization methods (section 2.1, section 2.5), as well as optimization methods (section 2.2) to obtain better skeletons. The skeleton (centreline body) contains the same posture information as the whole body (segmentation), with the advantage that the skeleton reduces the information and makes it easier to identify the location of the head, body, and tail, unlike using segmentations where overlapping parts of the same worm (coiling) or overlapping parts of other worms or plate noise (aggregations) can be confused. In other cases, where the worm is not connected to itself or others, segmentation would be a good tool to identify the head and tail, because the tail is thinner than the head.

The method proposed in paper 2 (section 2.2) to predict the next pose in aggregation cases, consisted of generating all possible pose solutions and using an optimizer to obtain the best one. The optimizer used different evaluation criteria to obtain the best answer. To select the best criterion, different combinations were made between criteria proposed by other authors, overlapping with the previous pose, length, smoothness of the skeleton, and noise, and two new criteria were proposed, completeness and color. The best criteria were overlapping with the previous posture, completeness, and color which obtained a value of 0.70 with the IoU metric using the improved skeletonization method. Although an improvement of 2.66% was obtained with the IoU metric using the improved skeletonization method compared to a traditional skeletonization method, evaluating all possible solutions required a lot of processing time. To speed up this time and reduce the number of possible solutions, paper 5 (section 2.5) proposed to use the output images of a trained deep learning model to predict poses. These output images had missing pixels, but despite these problems, the predictions were sufficient to filter out bad solutions and speed up the optimization process by 77%.

Detection is a very important step and should be performed before the posture-tracking of *C. elegans*. Recognizing objects of interest and noise, i.e. detecting correctly, reduces the time of subsequent image processing. However, in low-resolution images, this is a challenge, the dynamic range of *C. elegans* colors in greyscale are low, and in some cases it blends in with other elements of the environment. Paper 1 (section 2.1), on the one hand, proposed to use traditional image processing techniques to detect *C. elegans* and filter out noise. For this, the image sequence was processed to detect moving objects and static objects. Length and shape criteria were then used to verify that the moving objects were live *C. elegans*. All this processing had good detection of *C. elegans*, but the computational cost was high. Paper 3 (section 2.3), on the other hand, used two deep learning models, YoloV5 and Fast R-CNN, to detect *C. elegans* in whole Petri dish images, the detection results were high, 0.957 and 0.947 respectively, while the computational cost was low, 6.03[ms] and 43.25[ms] respectively, but in special cases the models mistook environmental noise for *C. elegans*. It is worth mentioning that the detection was performed on static images, the results with these deep learning models could be further improved if performed on image sequences.

All methods and algorithms in this thesis were implemented on low-resolution images. A low-resolution imaging system (SIVIS [67]) was used, because these are cheaper, more flexible, and the computational and memory cost are low compared to high-resolution imaging systems, not to mention that the system used is open source hardware and software. This system was used in other works and proved to be sufficient to automate *C. elegans* assays such as lifespan, healthspan, motility, etc. Another advantage of the images obtained with this low-resolution system is that it allowed the development of a synthetic imaging simulator of whole Petri dishes. This simulator was able to generate individual, coiled, parallel aggregation and cross behaviours of *C. elegans* (sections 2.4, 2.5). This was possible because at low-resolution the color configuration of the *C. elegans* is also limited. When overlaps occur, the colors are preserved, i.e. the colors do not merge or change. This simulator allowed us to generate a customized dataset to train deep learning models and solve cases of aggregations, obtaining faster results in posture-tracking compared to traditional methods.

Chapter 4

Conclussions and future research

Automated computer vision techniques have proven to be very powerful tools to help automate manual processes and visual tasks performed by humans, such as monitoring, detecting, and postures-tracking of textitCaenorhabditis elegans (C. elegans). In addition to being able to obtain different metrics more accurately and faster than human operators.

Posture-tracking and behavioural analysis of *C. elegans* is a difficult task, especially in cases of aggregation, where it may be impossible to find a solution. In this thesis, new automatic posture detection and prediction methods based on skeletonization methods were developed which helped the main objective of this thesis. Using the skeletons (centreline of the *C. elegans*) allowed us to better identify the postures of each individual in different cases of aggregation. It also helped to avoid changes and loss of identity in low-resolution image sequences. For this reason, and because of its simplicity (reduced body shape), we can say that the skeleton is a powerful tool for posture-tracking of *C. elegans* as well as for behavioural analysis.

The *C. elegans* posture-tracking methods presented were developed in the area of computer vision research using traditional image processing techniques as well as deep learning techniques, both on low-resolution images. Both techniques have their advantages and disadvantages, however, combining both allowed for more accurate and faster results. Paper 5 (Section 2.5), compared the proposed tracking method with both techniques and the proposed tracking method with traditional techniques (Section 2.2). The results showed an average improvement of 2.34% using IoU values, a 75% reduction in computational cost (Section 2.5), and a 16.96% improvement using the MOTA metric (Section A.4).

4.1 Future research

Extend the research of this thesis to other image capture devices, for example, the Multi-view cartesian robot prototype [68] or other prototypes that are currently being developed in the artificial vision and robotics research group of the Institute of Automatics and Industrial Informatics (ai2) at the Universitat Politècnica de València. This will allow the development of more automatic methods for studying behaviour in drug tests and other experiments with *C. elegans*.

New methods of skeletonization and posture prediction can also be developed based on the research performed in this thesis. This can be used to analyze different behaviours of different strains of *C. elegans* using different resolution images, image sequences, and fluorescent images.

Finally, the low-resolution synthetic imaging simulator proposed in this thesis opens a line of research in synthetic-real imaging. The simulator can be improved to generate synthetic images with more types of behaviours related to pathologies and neurodegenerative diseases. This will allow to extend the prediction and tracking of postures of other *C. elegans* strains by automating different types of healtspan assays. The simulator can also be used to train more complex deep learning models and improve the results obtained with the methods proposed in this thesis.

Appendix A

Supplementary information

a.1 Paper1

Improving skeleton algorithm for helping *Caenorhabditis elegans* trackers

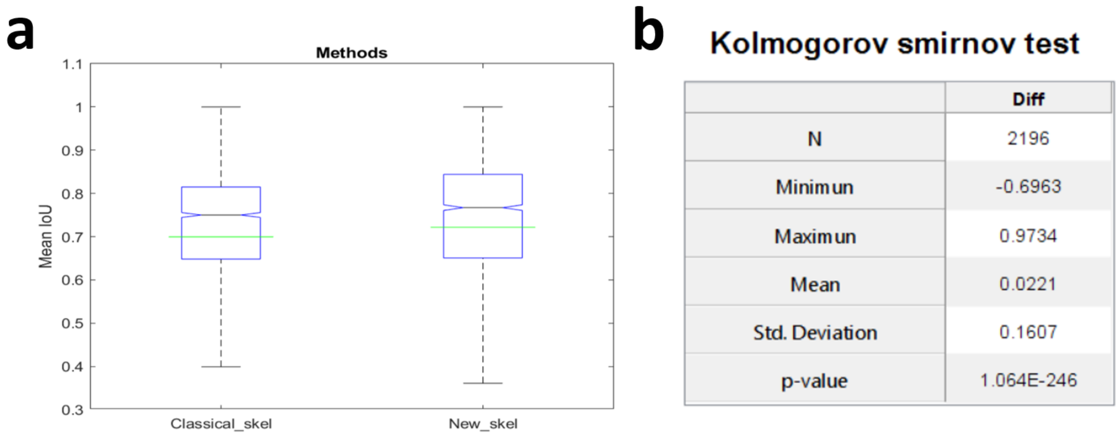
Pablo E. Layana Castro[1], **Joan Carles Puchalt**[1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Layana Castro, P. E., Puchalt, J. C., and Sánchez-Salmerón, A. J. (2020). Improving skeleton algorithm for helping *Caenorhabditis elegans* trackers. Scientific Reports, 10(1), 22247. https://doi.org/10.1038/s41598-020-79430-8.

Summary table IoU index1

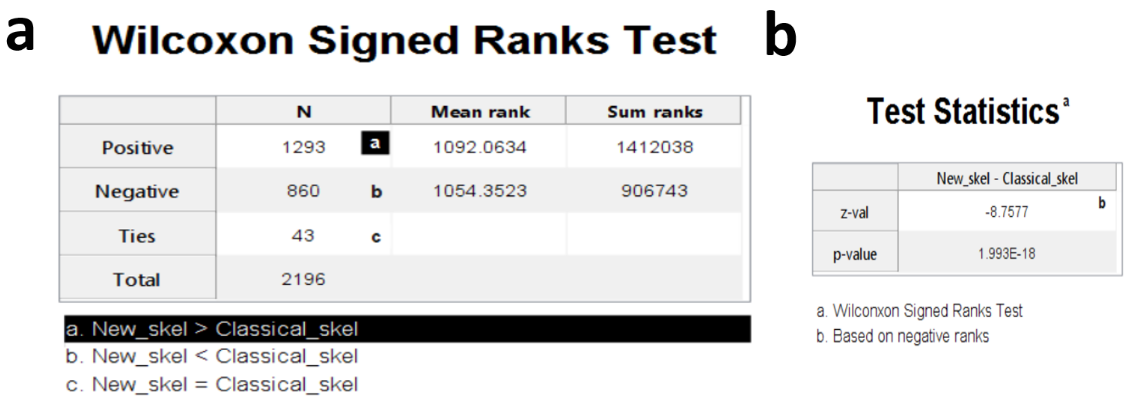
			Mean IoU		Standard Deviation		Results %
Problematic cases	Total tracks	Total pose	New skel	Classical skel	New skel	Classical skel	Improvement
Self-occluded	212	803	0,77	0,76	0,14	0,13	0,91
Noise contact	17	509	0,68	0,66	0,22	0,23	2,07
Partial bodies aggregation	53	828	0,7	0,68	0,18	0,17	2,25
Full bodies aggregation	4	56	0,69	0,47	0,2	0,26	21,5

Table A.1: **Summary of the comparison of automatic methods with the first IoU index.** This table shows the results obtained from all the experiments carried out and the improvements for each behavior using the first IoU evaluation method, the evaluated values indicate the percentage of success with respect to the reconstructed body of the skeleton labeled manually. (see experiments and results).



	Diff
N	2196
Minimun	-0.6963
Maximun	0.9734
Mean	0.0221
Std. Deviation	0.1607
p-value	1.064E-246

Figure A.1: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 2196, mean = 0.7216, median = 0.7671, std. deviation = 0.1805, variance = 0.0326. Classical_skel, N = 2196, mean = 0.6996, median = 0.75, std. deviation = 0.1839, variance = 0.0338. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 1.06E-246 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	1293	a	1092.0634	1412038
Negative	860	b	1054.3523	906743
Ties	43	c		
Total	2196			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

Test Statistics[a]

	New_skel - Classical_skel
z-val	-8.7577 b
p-value	1.993E-18

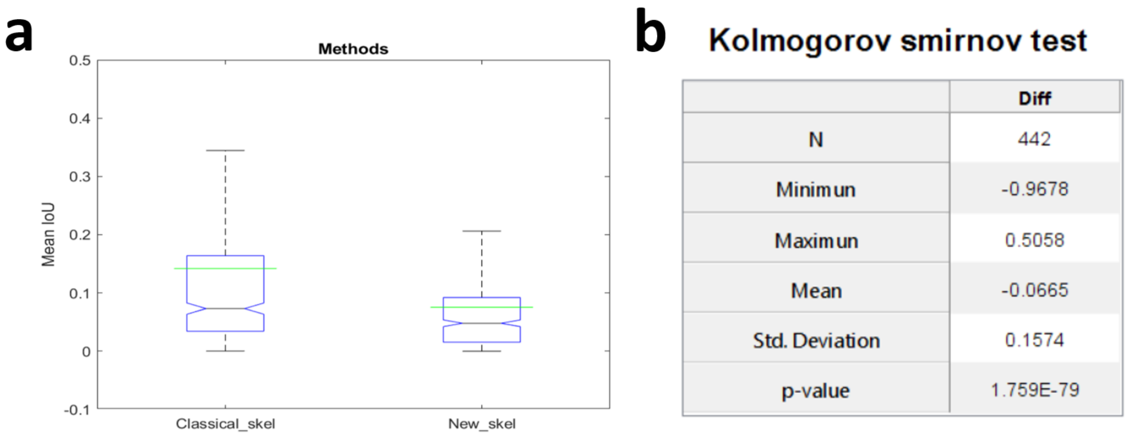
a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.2: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 1.99E-18 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

Summary table IoU index2

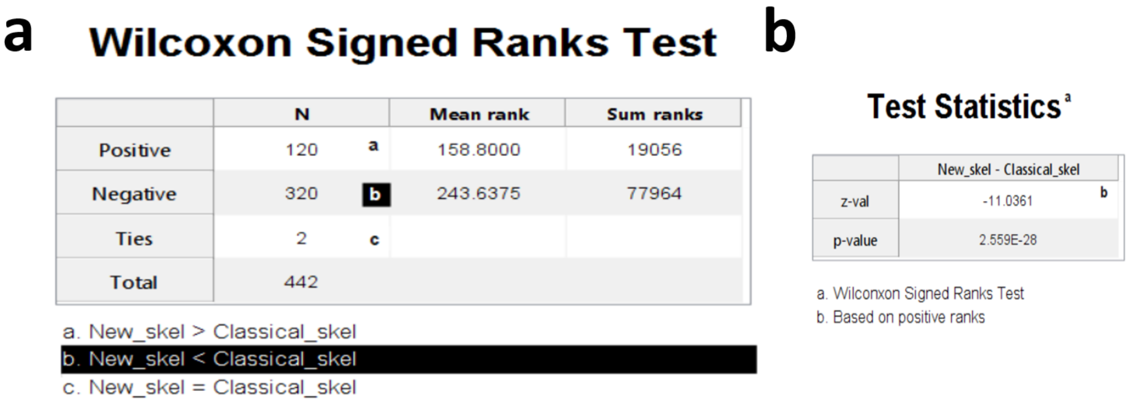
			Mean IoU		Standard Deviation		Results %
Problematic cases	Total tracks	Total pose	New skel	Classical skel	New skel	Classical skel	Improvement
Partial bodies aggregation	53	414	0,07	0,1	0,08	0,11	3,83
Full bodies aggregation	4	28	0,21	0,69	0,2	0,15	48,33

Table A.2: **Summary of the comparison of automatic methods with the second IoU index.** This table shows the results obtained from all the experiments carried out and the improvements for each behavior using the second IoU evaluation method, the evaluated values indicate the percentage of how connected they are to each other (see experiments and results).



	Diff
N	442
Minimun	-0.9678
Maximun	0.5058
Mean	-0.0665
Std. Deviation	0.1574
p-value	1.759E-79

Figure A.3: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 442, mean = 0.0754, median = 0.0479, std. deviation = 0.0966, variance = 0.0093. Classical_skel, N = 442, mean = 0.1419, median = 0.0732, std. deviation = 0.1826, variance = 0.0333. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 1.76E-79 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	120	a	158.8000	19056
Negative	320	b	243.6375	77964
Ties	2	c		
Total	442			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

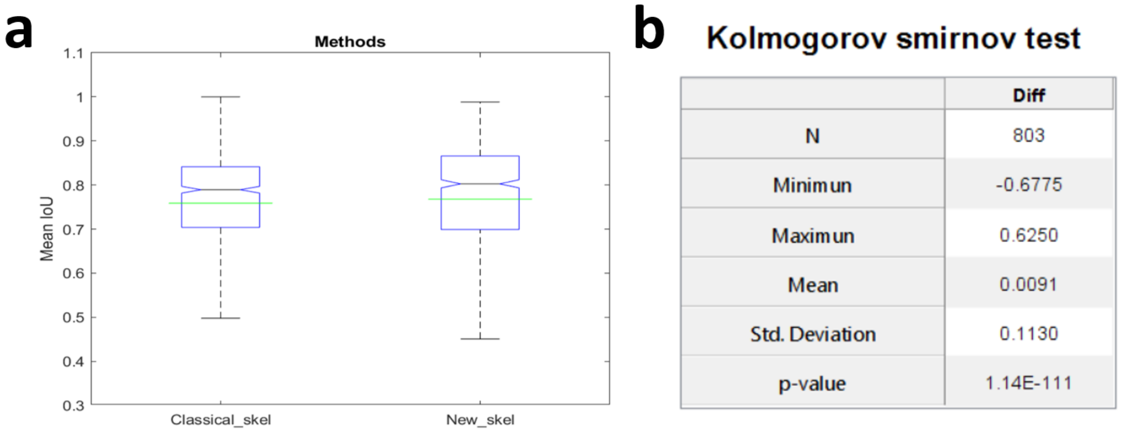
Test Statistics[a]

	New_skel - Classical_skel
z-val	-11.0361[b]
p-value	2.559E-28

a. Wilconxon Signed Ranks Test
b. Based on positive ranks

Figure A.4: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 2.56E-28 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

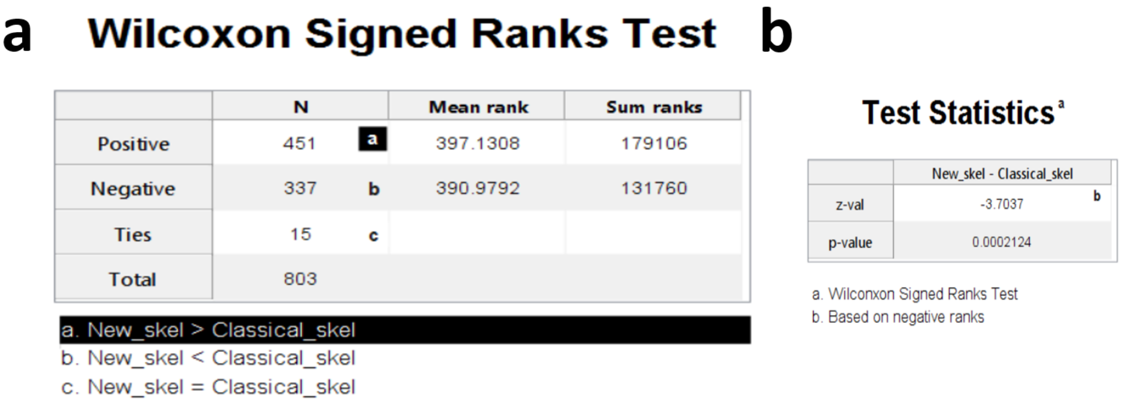
Self occluded index1



Kolmogorov smirnov test

	Diff
N	803
Minimun	-0.6775
Maximun	0.6250
Mean	0.0091
Std. Deviation	0.1130
p-value	1.14E-111

Figure A.5: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 803, mean = 0.7678, median = 0.8026, std. deviation = 0.1373, variance = 0.0188. Classical_skel, N = 803, mean = 0.7588, median = 0.7892, std. deviation = 0.1290, variance = 0.0166. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 1.14E-111 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	451	a	397.1308	179106
Negative	337	b	390.9792	131760
Ties	15	c		
Total	803			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

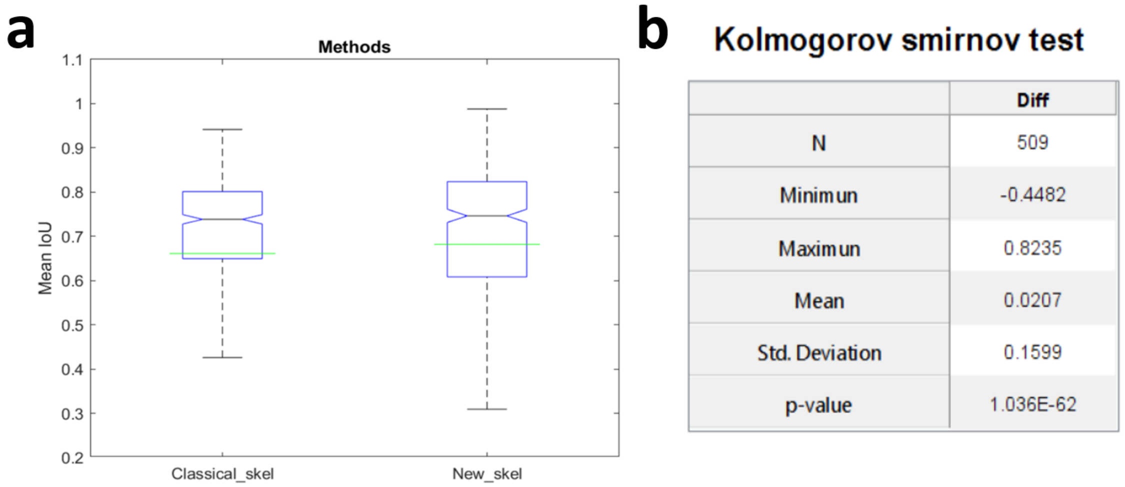
Test Statistics[a]

	New_skel - Classical_skel
z-val	-3.7037[b]
p-value	0.0002124

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.6: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 2.12E-4 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

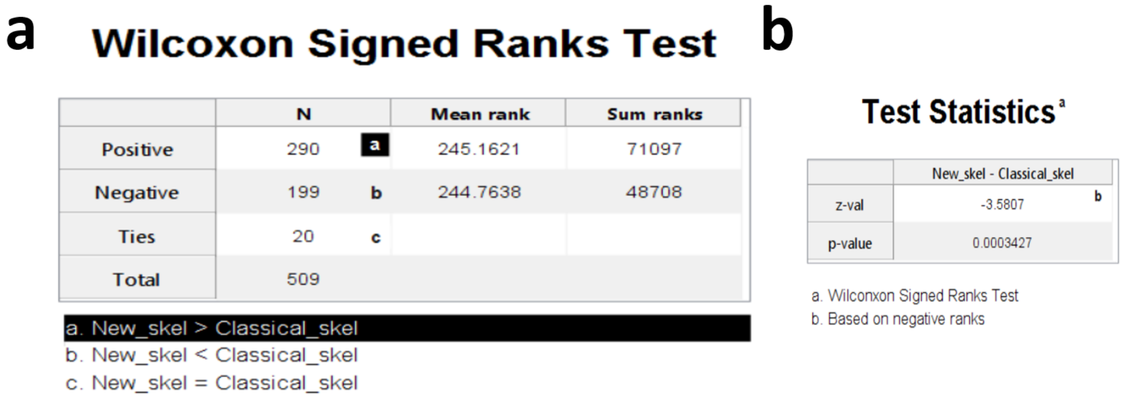
Noise contact index1



Kolmogorov smirnov test

	Diff
N	509
Minimun	-0.4482
Maximun	0.8235
Mean	0.0207
Std. Deviation	0.1599
p-value	1.036E-62

Figure A.7: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 509, mean = 0.6818, median = 0.7463, std. deviation = 0.2152, variance = 0.0463. Classical_skel, N = 509, mean = 0.6611, median = 0.7384, std. deviation = 0.2300, variance = 0.0529. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 1.04E-62 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	290	a	245.1621	71097
Negative	199	b	244.7638	48708
Ties	20	c		
Total	509			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

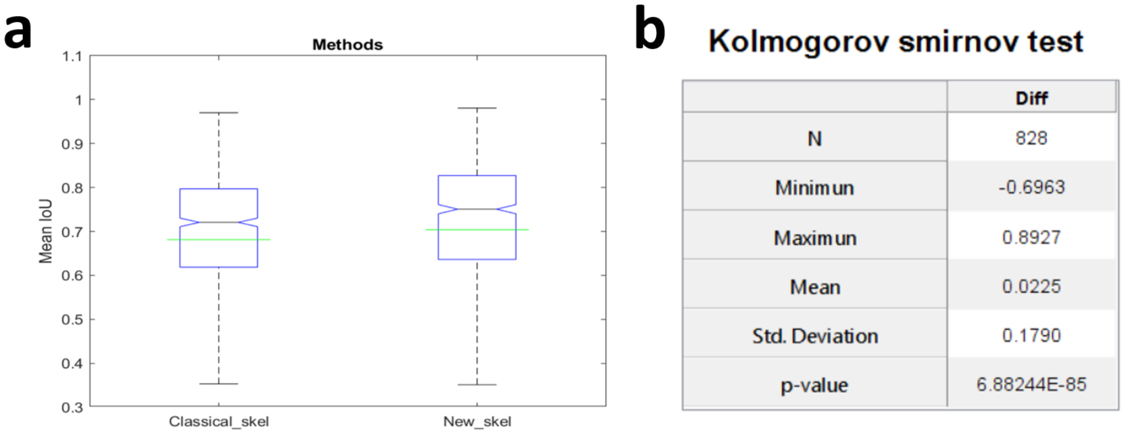
Test Statistics[a]

	New_skel - Classical_skel
z-val	-3.5807 [b]
p-value	0.0003427

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.8: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 3.43E-4 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

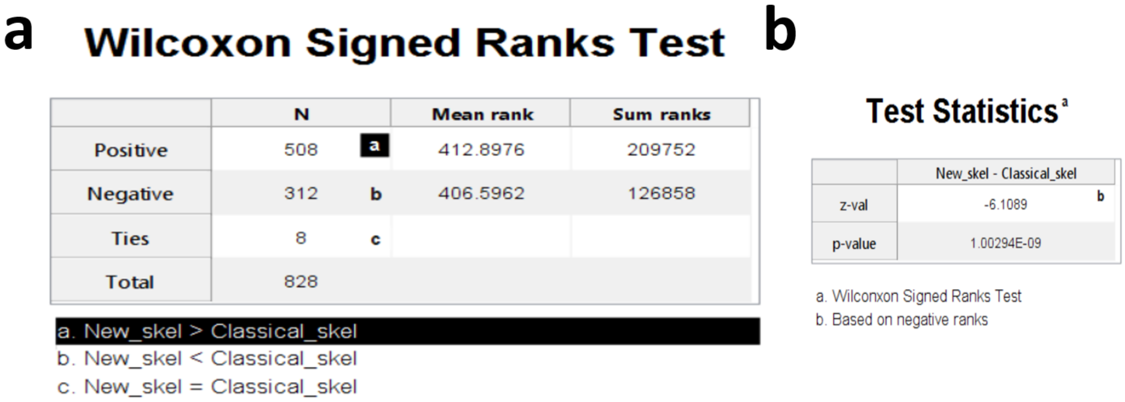
Partial bodies aggregation index1



Kolmogorov smirnov test

	Diff
N	828
Minimun	-0.6963
Maximun	0.8927
Mean	0.0225
Std. Deviation	0.1790
p-value	6.88244E-85

Figure A.9: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 828, mean = 0.7038, median = 0.7506, std. deviation = 0.1837, variance = 0.0337. Classical_skel, N = 828, mean = 0.6813, median = 0.7206, std. deviation = 0.1688, variance = 0.0285. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 6.88E-85 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	508	a	412.8976	209752
Negative	312	b	406.5962	126858
Ties	8	c		
Total	828			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

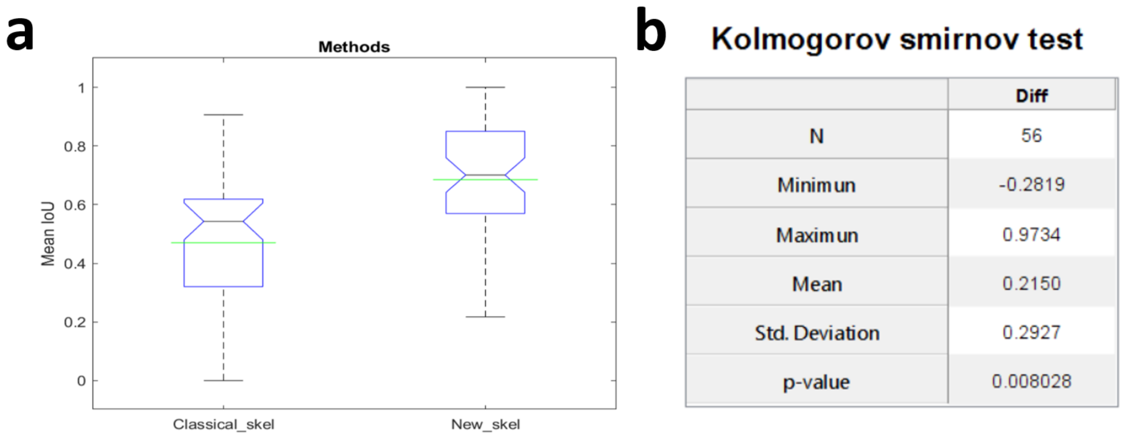
Test Statistics[a]

	New_skel - Classical_skel
z-val	-6.1089[b]
p-value	1.00294E-09

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.10: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 1E-9 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

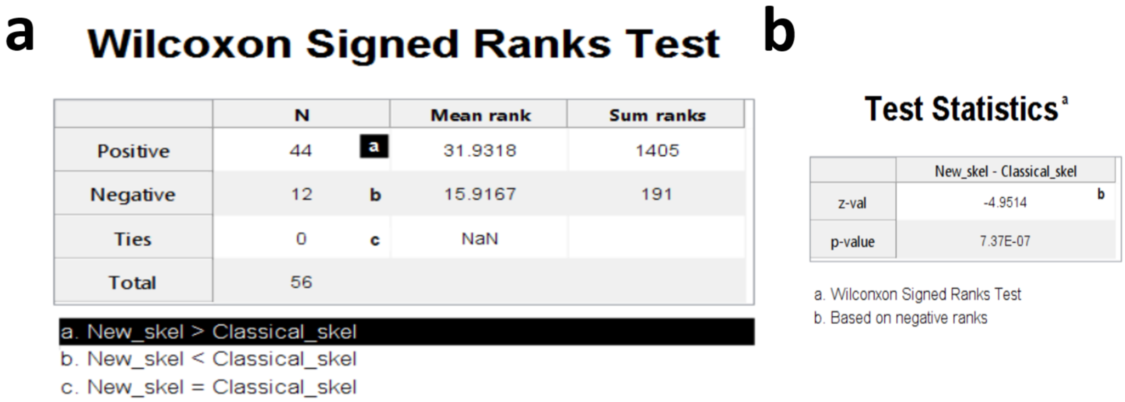
Full bodies aggregation index1



Kolmogorov smirnov test

	Diff
N	56
Minimun	-0.2819
Maximun	0.9734
Mean	0.2150
Std. Deviation	0.2927
p-value	0.008028

Figure A.11: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 56, mean = 0.6852, median = 0.7009, std. deviation = 0.1969, variance = 0.0388. Classical_skel, N = 56, mean = 0.4702, median = 0.5430, std. deviation = 0.2571, variance = 0.0661. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 8.03E-3 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	44	a	31.9318	1405
Negative	12	b	15.9167	191
Ties	0	c	NaN	
Total	56			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

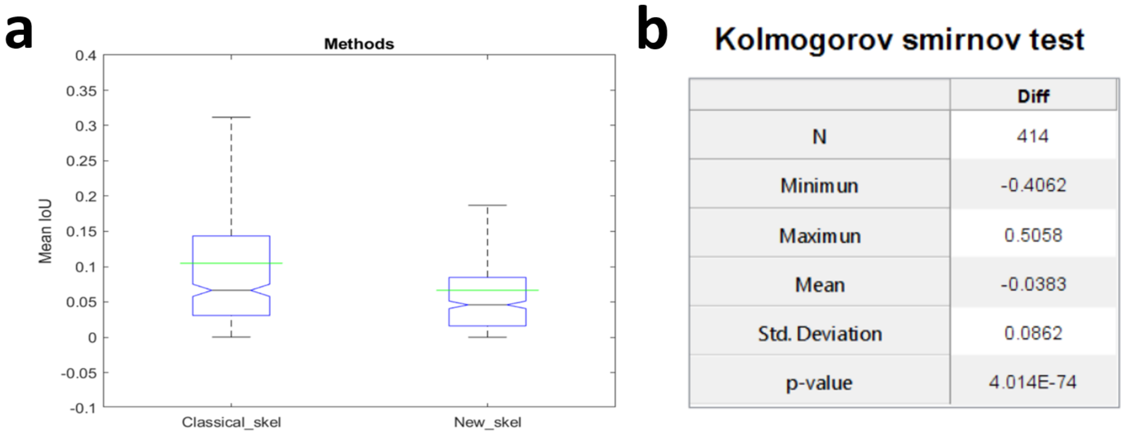
Test Statistics[a]

	New_skel - Classical_skel
z-val	-4.9514 [b]
p-value	7.37E-07

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.12: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 7.37E-7 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

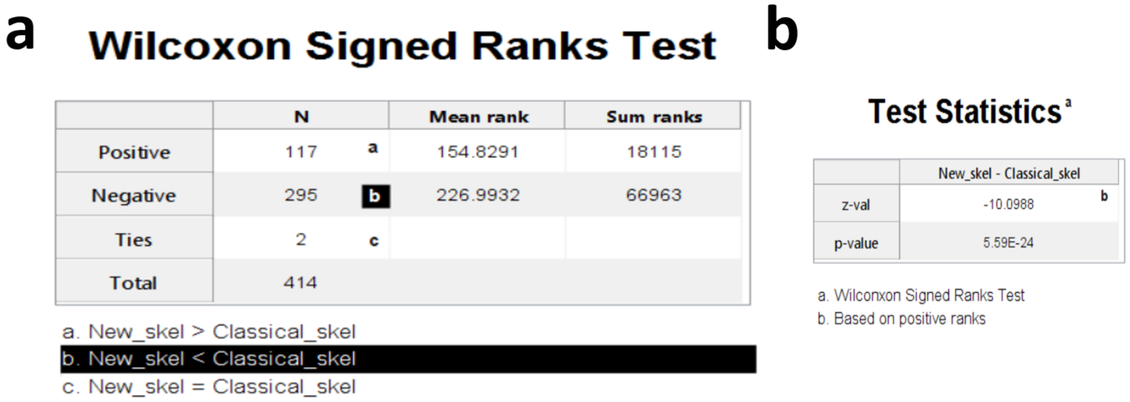
Partial bodies aggregation index2



Kolmogorov smirnov test

	Diff
N	414
Minimun	-0.4062
Maximun	0.5058
Mean	-0.0383
Std. Deviation	0.0862
p-value	4.014E-74

Figure A.13: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 414, mean = 0.0665, median = 0.0459, std. deviation = 0.0788, variance = 0.0062. Classical_skel, N = 414, mean = 0.1049, median = 0.0665, std. deviation = 0.1109, variance = 0.0123. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 4.01E-74 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	117	a	154.8291	18115
Negative	295	b	226.9932	66963
Ties	2	c		
Total	414			

a. New_skel > Classical_skel
b. New_skel < Classical_skel
c. New_skel = Classical_skel

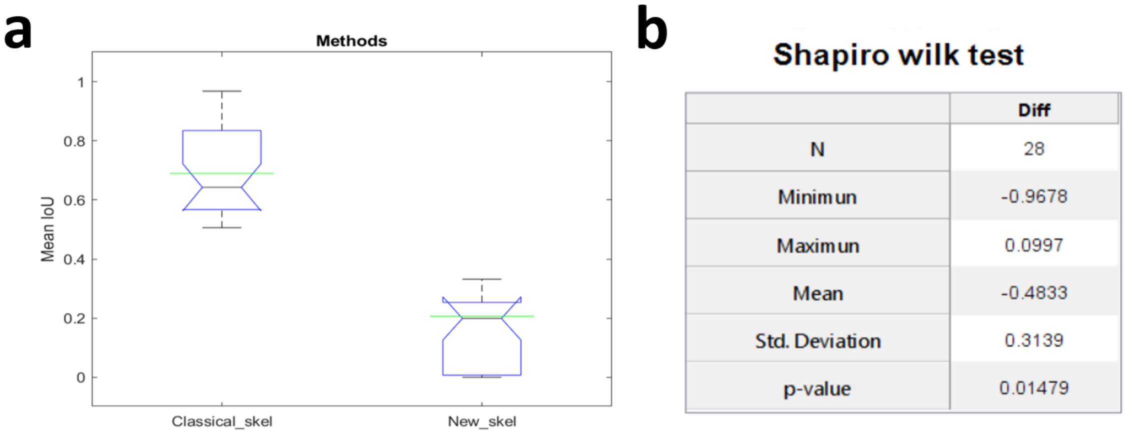
Test Statistics[a]

	New_skel - Classical_skel
z-val	-10.0988 [b]
p-value	5.59E-24

a. Wilconxon Signed Ranks Test
b. Based on positive ranks

Figure A.14: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with wilcoxon rank test was 5.59E-24 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

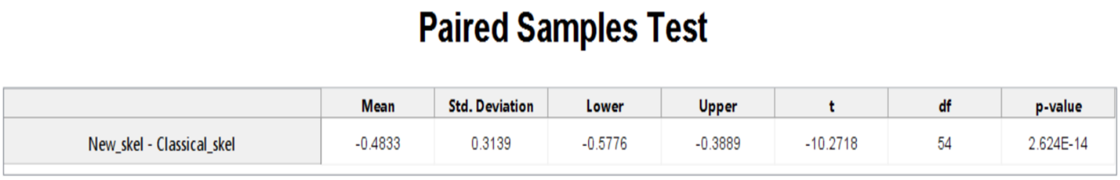
Full bodies aggregation index2



Shapiro wilk test

	Diff
N	28
Minimun	-0.9678
Maximun	0.0997
Mean	-0.4833
Std. Deviation	0.3139
p-value	0.01479

Figure A.15: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. New_skel, N = 28, mean = 0.2066, median = 0.1993, std. deviation = 0.1958, variance = 0.0383. Classical_skel, N = 28, mean = 0.6899, median = 0.6426, std. deviation = 0.1538, variance = 0.0236. (**b**) Normality test on the difference of methods (new - classical). The p-value obtained was 1.48E-2 less than the significance value of 0.05, so the null hypothesis was accepted (data come from normal distribution). Once the null hypothesis was accepted, t-student test was used to evaluate both methods.



Paired Samples Test

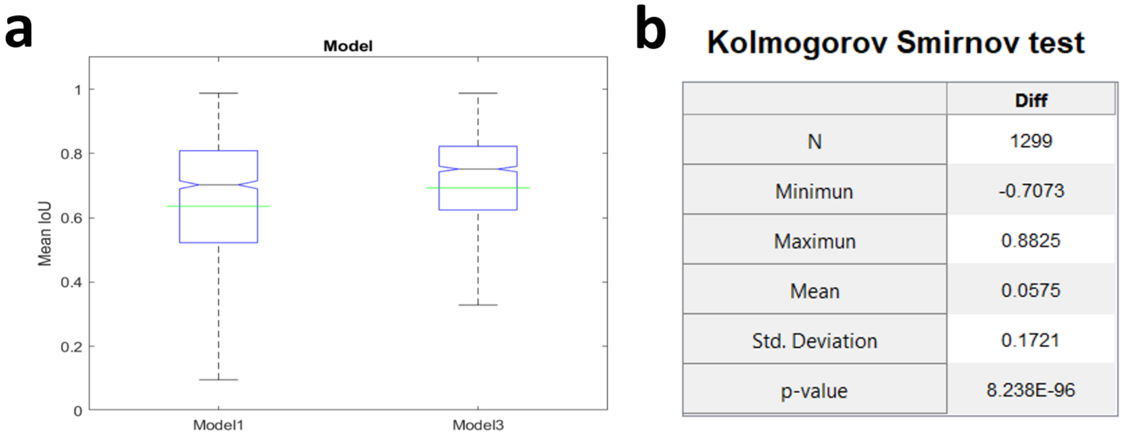
	Mean	Std. Deviation	Lower	Upper	t	df	p-value
New_skel - Classical_skel	-0.4833	0.3139	-0.5776	-0.3889	-10.2718	54	2.624E-14

Figure A.16: **T-student paired samples test.** T-student paired samples test table shows the difference that exists in 2 related samples. P-value obtained was 2.62E-14 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

a.2 Paper2

a.2.1 Apendix A

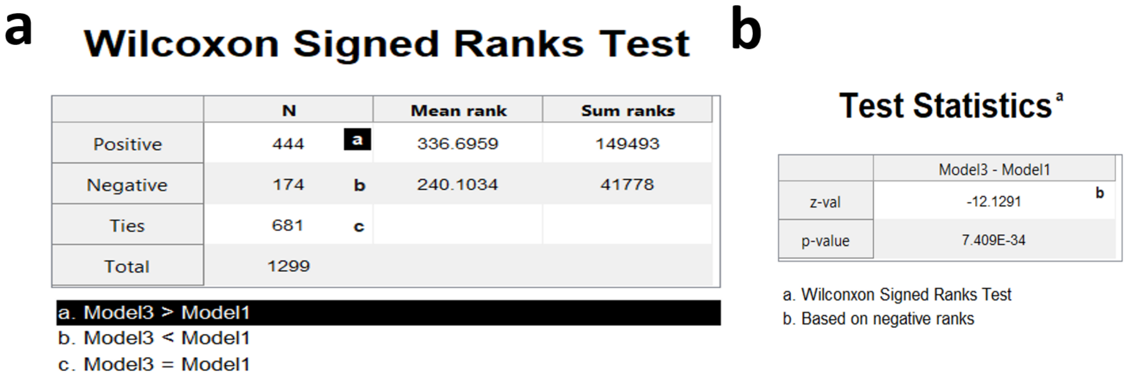
IoU comparison in models 3 and 1



Kolmogorov Smirnov test

	Diff
N	1299
Minimun	-0.7073
Maximun	0.8825
Mean	0.0575
Std. Deviation	0.1721
p-value	8.238E-96

Figure A.17: **Box plot and normality test of the difference of both models.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. Model1, N = 1299, mean = 0.6358, median = 0.7027, std. deviation = 0.2289, variance = 0.0540. Model3, N = 1299, mean = 0.6933, median = 0.7517, std. deviation = 0.1940, variance = 0.0376. (**b**) Normality test on the difference of methods (Model3 - Model1). The p-value obtained was 7.41E34 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	444	a	336.6959	149493
Negative	174	b	240.1034	41778
Ties	681	c		
Total	1299			

a. Model3 > Model1
b. Model3 < Model1
c. Model3 = Model1

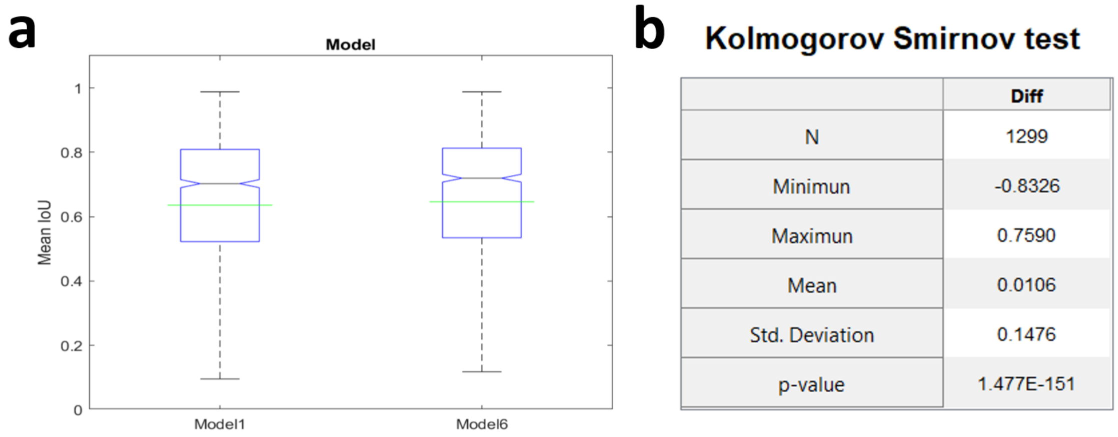
Test Statistics[a]

	Model3 - Model1
z-val	-12.1291 [b]
p-value	7.409E-34

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.18: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with Wilcoxon rank test was 741E-34 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both models.

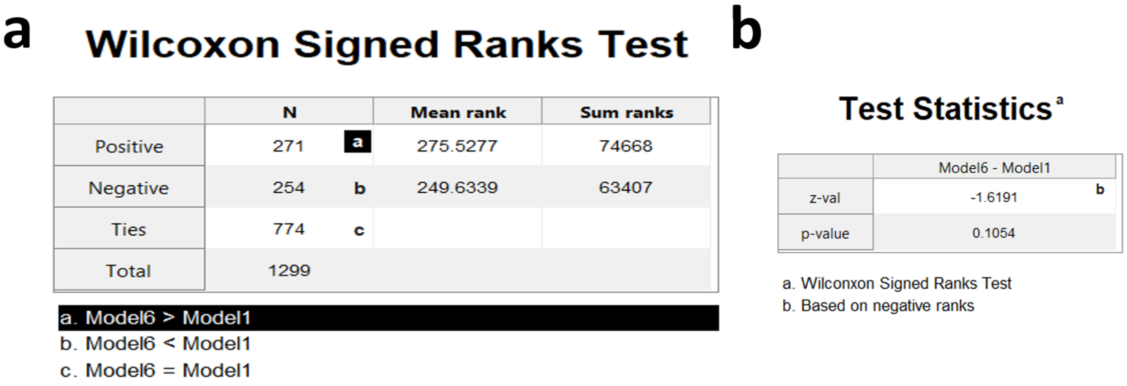
IoU comparison in models 6 and 1



Kolmogorov Smirnov test

	Diff
N	1299
Minimun	-0.8326
Maximun	0.7590
Mean	0.0106
Std. Deviation	0.1476
p-value	1.477E-151

Figure A.19: **Box plot and normality test of the difference of both models.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median.Model1, N = 1299, mean = 0.6358, median = 0.7027, std. deviation = 0.2289, variance = 0.0540. Model6, N = 1299, mean = 0.6464, median = 0.7197, std. deviation = 0.2300, variance = 0.0529. (**b**) Normality test on the difference of methods (Model6 - Model1). The p-value obtained was 1.48E-151 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	271	a	275.5277	74668
Negative	254	b	249.6339	63407
Ties	774	c		
Total	1299			

a. Model6 > Model1
b. Model6 < Model1
c. Model6 = Model1

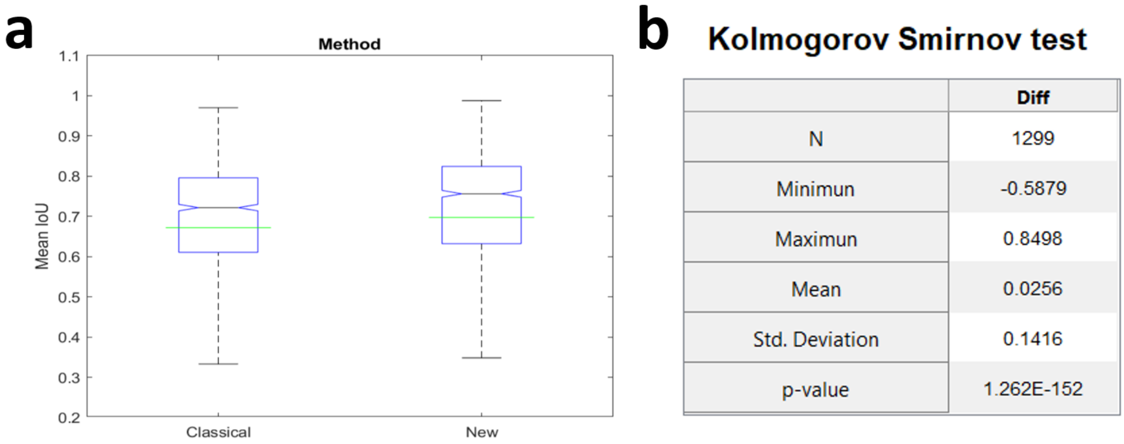
Test Statistics[a]

	Model6 - Model1	
z-val	-1.6191	b
p-value	0.1054	

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.20: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with Wilcoxon rank test was 0.1054 less than the significance value of 0.05, so it was concluded that there wasn't a statistically significant difference between both models.

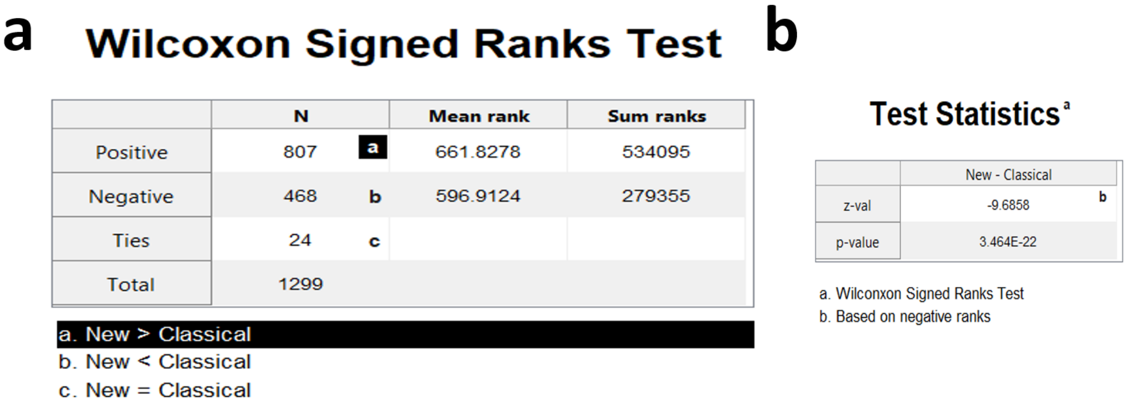
IoU comparison New and Classical method (model7)



Kolmogorov Smirnov test

	Diff
N	1299
Minimun	-0.5879
Maximun	0.8498
Mean	0.0256
Std. Deviation	0.1416
p-value	1.262E-152

Figure A.21: **Box plot and normality test of the difference of both methods.** (**a**) Box plot, green line indicates the mean in both graphs, and gray line indicates the median. Classical method, N = 1299, mean = 0.6719, median = 0.7218, std. deviation = 0.1842, variance = 0.0339. New method, N = 1299, mean = 0.6975, median = 0.7562, std. deviation = 0.1929, variance = 0.0372. (**b**) Normality test on the difference of methods (New - classical). The p-value obtained was 1.26E-152 less than the significance value of 0.05, so the null hypothesis was rejected and the alternative hypothesis H1 was accepted (data did not come from normal distribution). Once the alternative hypothesis was accepted, Wilcoxon signed ranks test was used to evaluate both methods.



Wilcoxon Signed Ranks Test

	N		Mean rank	Sum ranks
Positive	807	a	661.8278	534095
Negative	468	b	596.9124	279355
Ties	24	c		
Total	1299			

a. New > Classical
b. New < Classical
c. New = Classical

Test Statistics[a]

	New - Classical
z-val	-9.6858[b]
p-value	3.464E-22

a. Wilconxon Signed Ranks Test
b. Based on negative ranks

Figure A.22: **Wilcoxon signed rank test.** (**a**) The Wilcoxon signed rank test table shows the difference that exists in 2 related samples through positive, negative and tie ranges. (**b**) P-value obtained with Wilcoxon rank test was 3.46E-22 less than the significance value of 0.05, so it was concluded that there was a statistically significant difference between both methods.

a.2.2 Apendix B

Evaluation examples with model7

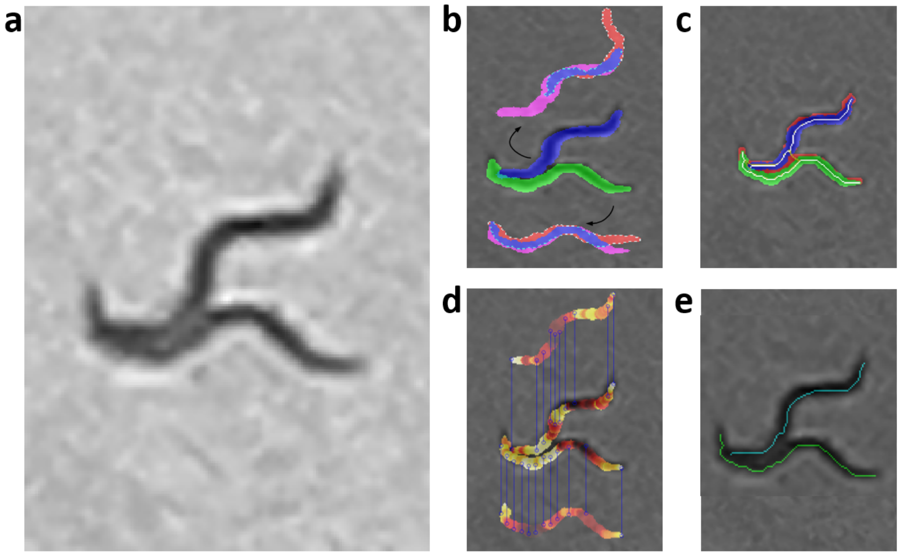


Figure A.23: **Model7 evaluation, example 1.** The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). (**a**) Grayscale image. (**b**) Overlap criterion evaluation. (**c**) Completeness criterion evaluation. (**d**) Color criterion evaluation. (**e**) Optimization result.

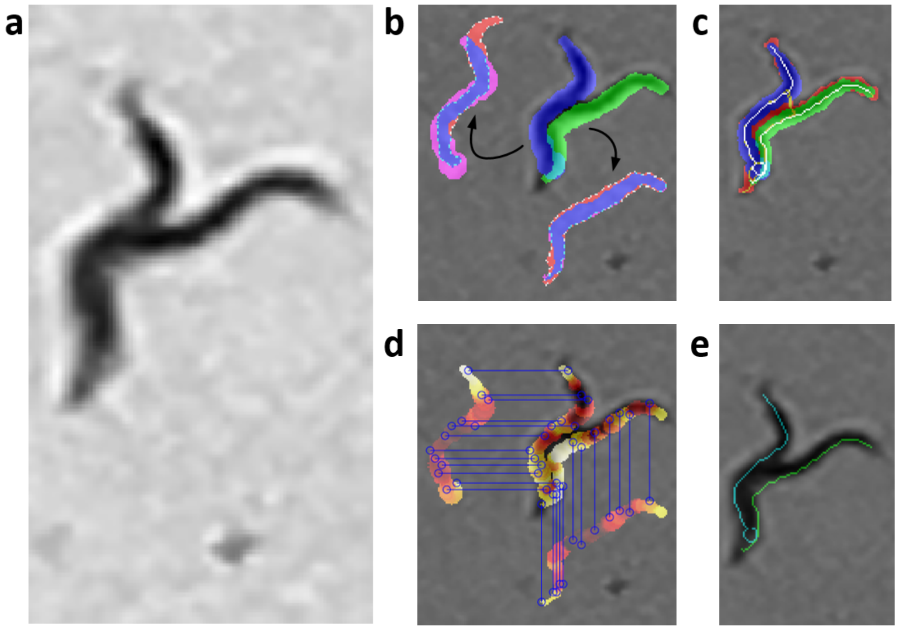


Figure A.24: **Model7 evaluation, example 2.** The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). (**a**) Grayscale image. (**b**) Overlap criterion evaluation. (**c**) Completeness criterion evaluation. (**d**) Color criterion evaluation. (**e**) Optimization result.

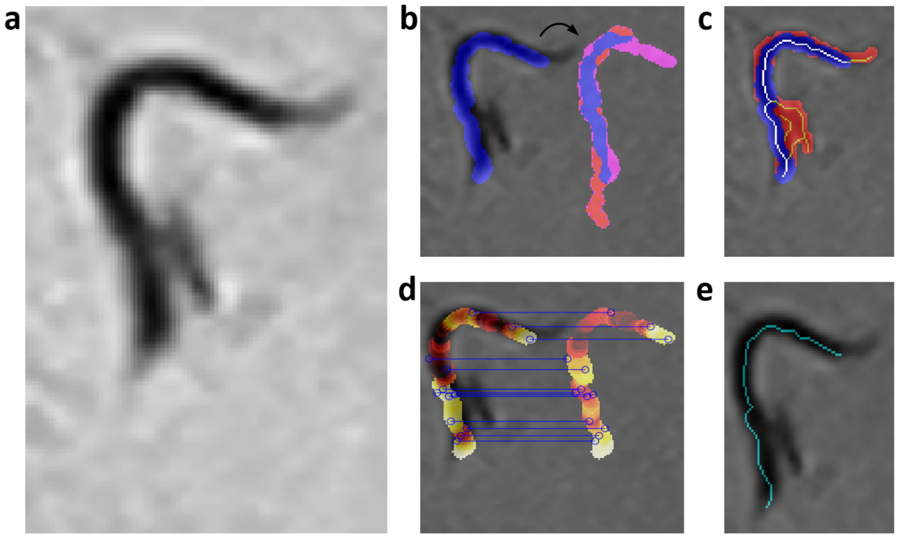


Figure A.25: **Model7 evaluation, example 3.** The yellow and white pixels are the resulting skeleton using the improved form of skeletonizing. The white pixels are the pixels of the skeleton prediction, which are used to reconstruct the body of each worm (segmentation in blue and green). (**a**) Grayscale image. (**b**) Overlap criterion evaluation. (**c**) Completeness criterion evaluation. (**d**) Color criterion evaluation. (**e**) Optimization result.

Failure cases

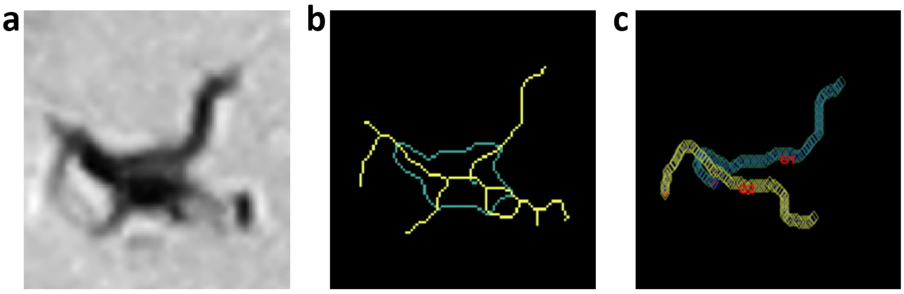


Figure A.26: **Model7 evaluation, example 4.** Errors occurred by the presence of noise. (**a**) Grayscale image. (**b**) Resulting skeleton using the improved form of skeletonizing. (**c**) Optimization result.

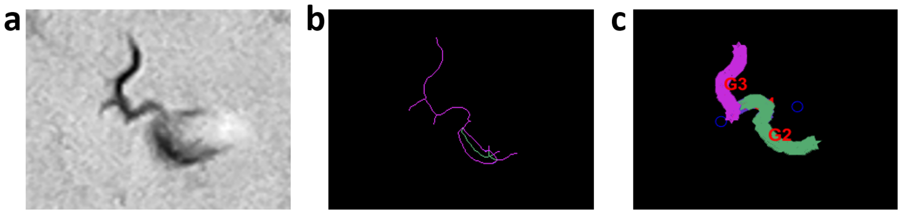


Figure A.27: **Model7 evaluation, example 5.** Errors occurred by the presence of noise. (**a**) Grayscale image. (**b**) Resulting skeleton using the improved form of skeletonizing (**c**) Optimization result.

a.3 Paper4

a.3.1 Appendix A

Other images

The synthetic image sequence was generated using two functions: Simulator and Create Image (Image generator). The first generated PTS and backgrounds files from pre-processing files (TXT), worm skeleton files (XML), and background images without worms. While the second function used these files to generate the images (Figure A.28).The PTS files contained skeleton point information, as well as color and width values for each skeleton pixel. The synthetic image generator was developed in C++ using the OpenCV tool for image processing and integrated into Python through the SWIG application.

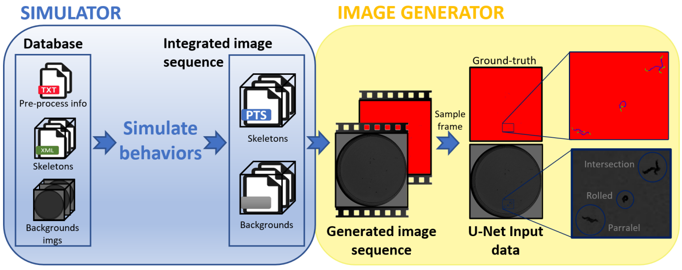
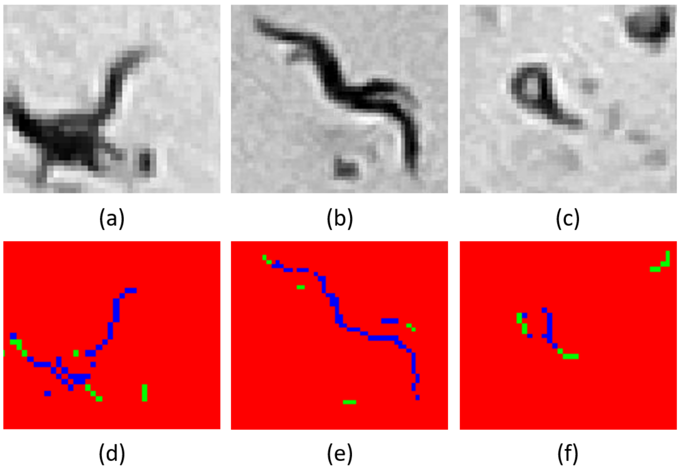


Figure A.28: **Pipeline code.**

Figure A.29 shows different skeleton errors. Figure A.29a, d show errors occurred by the presence of noise similar to the worm. Figure A.29b, e show errors when worms are highly aggregated, and Figure A.29c, f shows errors when worms are small and coiled upon itself. Figure A.30 show full image of Figure 2.41, and Figure A.31 show the comparison of previous work [41] with U-nets architectures using a box plot.


(a) (b) (c)

(d) (e) (f)

Figure A.29: **Error example of UMF U-Net model. (a)** Grayscale image of worm aggregated with noise. **(b)** Grayscale image of two worms aggregated at one end and part of the body. **(c)** Gray image of worm coiled upon itself. **(d, e, f)** Pixel encoding using the maximum value of the RGB channels. Red pixels are background pixels, blue pixels are worm body pixels, and green pixels are worm-end pixels. The results obtained with the rest of the models are similar to these.

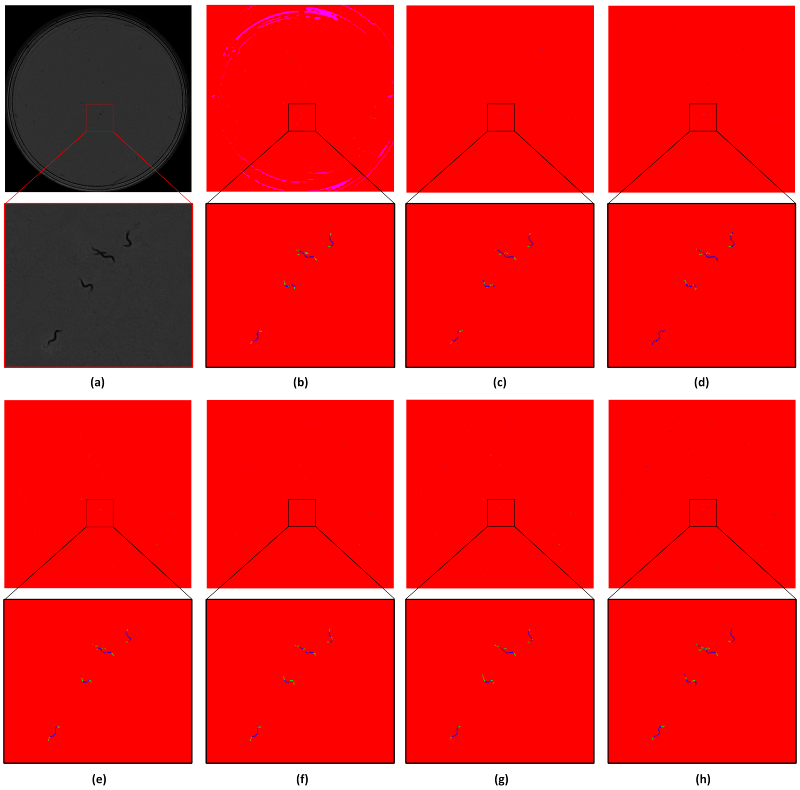


Figure A.30: **Comparison of skeletons obtained with the different U-Net architectures.** Full image of Figure 2.41. Each image presents a zoom to a certain area. **(a)** Grayscale image. **(b)** Result with standard U-Net [79], **(c)** Result with Alexandre's U-Net [1], **(d)** Result with UMF U-Net [63], **(e, f, g, h)** Result with SmaAt U-Net [93] (SmaAT Ds, SmaATDs At, SmaATDs At 4CBAMs and SmaAT respectively).

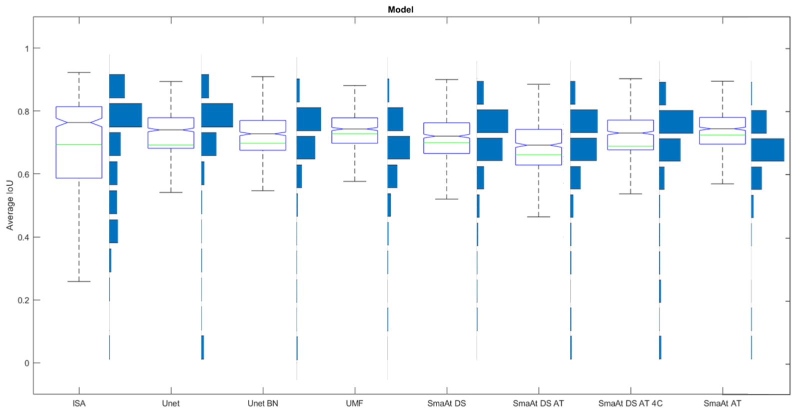


Figure A.31: **Comparison of previous work [41] with U-Nets architectures.** Green line indicates the mean in both graphs and gray line indicates the median. ISA N = 664, mean = 0.6936, median = 0.7635, standard deviation = 0.1649, variance = 0.0272. U-Net N = 664, mean = 0.6923, median = 0.7400, standard deviation = 0.1757, variance = 0.0309. U-Net A. N = 664, mean = 0.6977, median = 0.7276, standard deviation = 0.1378, variance = 0.0190. UMF U-Net N = 664, mean = 0.7279, median = 0.7430, standard deviation = 0.0871, variance = 0.0076. SmaAT DS N = 664, mean = 0.6993, median = 0.7201, standard deviation = 0.1135, variance = 0.0129. SmaAT DS AT N = 664, mean = 0.6613, median = 0.6916, standard deviation = 0.1407, variance = 0.0198. SmaAT DS AT 4C N = 664, mean = 0.6884, median = 0.7302, standard deviation = 0.1646, variance = 0.0271. SmaAT AT N = 664, mean = 0.7240, median = 0.7436, standard deviation = 0.1080, variance = 0.0117.

a.3.2 Appendix B

Analysis results tables

Table A.3, Table A.4, Table A.5, show the average results from IoU and error pixel distance (E.D.) for each problematic case and for each skeleton segmentation class.

Table A.3: **Analysis of results by class (worm ends, worm body) for aggregation cases.**

	Avg. worm-ends class		Avg. worm-body class		Total average worm	
Model	IoU	E.D.	IoU	E.D.	IoU ± IC 95%	E.D.± IC 95%
ISA	0.6627	0.5787	0.7635	0.5604	0.7625±0.0065	0.5659±0.0206
U-Net	**0.6725**	**0.5634**	**0.768**	**0.5554**	**0.7634±0.0046**	**0.5551±0.0186**
U-Net A.	0.5391	0.8740	0.7027	0.6915	0.6858±0.0103	0.7822±0.0714
UMF U-Net	0.5848	0.6093	0.7400	0.6138	0.6992±0.0083	0.6090±0.0151
SmaAt DS	0.4558	0.5673	0.6552	0.5810	0.6339±0.0141	0.5737±0.0217
SmaAt DS AT	0.4343	0.5989	0.5819	0.5822	0.5759±0.0184	0.5870±0.0282
SmaAt DS AT 4C	0.4374	0.7422	0.5637	0.7512	0.5713±0.0219	0.7471±0.0394
SmaAt AT	0.5551	0.8472	0.7135	0.8385	0.6767±0.0107	0.8359±0.0430

Table A.4: **Analysis of results by class (worm ends, worm body) for aggregation with noise cases.**

	Avg. worm-ends class		Avg. worm-body class		Total average worm	
Model	IoU	E.D.	IoU	E.D.	IoU ± IC 95%	E.D.± IC 95%
ISA	0.5836	2.2960	0.7031	1.6793	0.6421±0.0137	2.1726±0.1731
U-Net	0.5650	0.9349	0.6678	0.8802	0.6510±0.0160	0.9001±0.0470
U-Net A.	0.5540	0.7804	0.7359	0.6232	0.6980±0.0116	0.6712±0.0335
UMF U-Net	0.5959	0.8476	0.7691	0.6358	0.7313±0.0053	0.7259±0.0412
SmaAt DS	0.5501	0.7772	0.7509	0.5992	0.7133±0.0080	0.6562±0.0332
SmaAt DS AT	0.5237	0.7426	0.7135	0.6125	0.6849±0.0085	0.6505±0.0199
SmaAt DS AT 4C	0.5908	0.8142	0.7493	0.6481	0.7193±0.0079	0.7032±0.0188
SmaAt AT	**0.6041**	**0.7312**	**0.7783**	**0.6097**	**0.7338±0.0049**	**0.6459±0.0382**

Table A.5: **Analysis of results by class (worm ends, worm body) for rolled cases.**

	Avg. worm-ends class		Avg. worm-body class		Total average worm	
Model	IoU	E.D.	IoU	E.D.	IoU ± IC 95%	E.D.± IC 95%
ISA	**0.7268**	**0.5685**	**0.8066**	**0.5083**	**0.8122±0.0040**	**0.554±0.0122**
U-Net	0.6796	0.6797	0.7507	0.6497	0.7600±0.0037	0.6678±0.0114
U-Net A.	0.6126	0.6267	0.7108	0.6027	0.7172±0.0097	0.6089±0.0164
UMF U-Net	0.6854	0.6142	0.7644	0.5680	0.7622±0.0048	0.5686±0.0124
SmaAt DS	0.6373	0.6190	0.7463	0.5762	0.7481±0.0046	0.5905±0.012
SmaAt DS AT	0.5845	0.6404	0.6897	0.6103	0.7004±0.0059	0.6150±0.0138
SmaAt DS AT 4C	0.6640	0.6369	0.7391	0.5911	0.7497±0.0044	0.6053±0.0164
SmaAt AT	0.6602	0.6038	0.7576	0.6173	0.7610±0.0077	0.6146±0.0125

a.4 Paper5

Automatic segmentation of *Caenorhabditis elegans* skeletons in worm aggregations using improved U-Net in low-resolution image sequences.

Pablo E. Layana Castro[1], **Antonio García Garví** [1], **Antonio-José Sánchez-Salmerón**[1,*]
[1] Universitat Politècnica de Valéncia, Instituto de Automática e Informática Industrial, Valencia, Spain

Layana Castro, P. E., Garví, A. G., Sánchez-Salmerón, A. J. (2023). Automatic segmentation of Caenorhabditis elegans skeletons in worm aggregations using improved U-Net in low-resolution image sequences. Heliyon. https://doi.org/10.1016/j.heliyon.2023.e14715.

MOTA metric

The poses used for the MOTA metric were all those of the worms from the beginning of the sequence to the end, 696 for aggregations between worms, and 463 for aggregations with noise. To obtain MOTA value (EquationA.1), the *FN*, *FP*, *IDS*, and *GT* parameters were used. *GT* was the total number of worms in the aggregation. *IDS* value was increased by one when the body of a predicted worm overlapped more with another worm than with its respective *GT*. For the overlap, the IoU value and a threshold of 0.5 were considered. On the other hand, when the Prediction did not coincide with the Ground-truth, the *FP* value increased by one, while the *FN* value increased by one when the Ground-truth did not coincide with the Prediction.

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDS_t}{\sum_t GT_t} \tag{A.1}$$

Table A.6: **Comparison of the values using the MOTA metric for 0.65 filter and 0 filter.** This table shows the values obtained using the MOTA metric with filter equal to 0.65 and without filter (previous works). The cases evaluated with the filter values were multi-worm aggregations (696 poses) and aggregations with noise (463 poses).

		MOTA	
Cases	Total pose	F=0.00	F=0.65
Aggregation	696	0.5546	**0.8190**
Noise	463	0.6147	**0.6330**
Total	1159	0.5777	**0.7473**

Image processing/results

The below images show examples of aggregation between worms and plate noise, Supplementary Figure A.32, A.33, A.34, A.35. To predict the next pose (skeleton) of each worm the proposed neural network model needs three images, the worm skeleton in the previous image (t-1), the previous gray image (t-1), and the current image (t). The result of each worm (network output) is used for a post-processing operation where an optimizing algorithm obtains the best result for each worm.

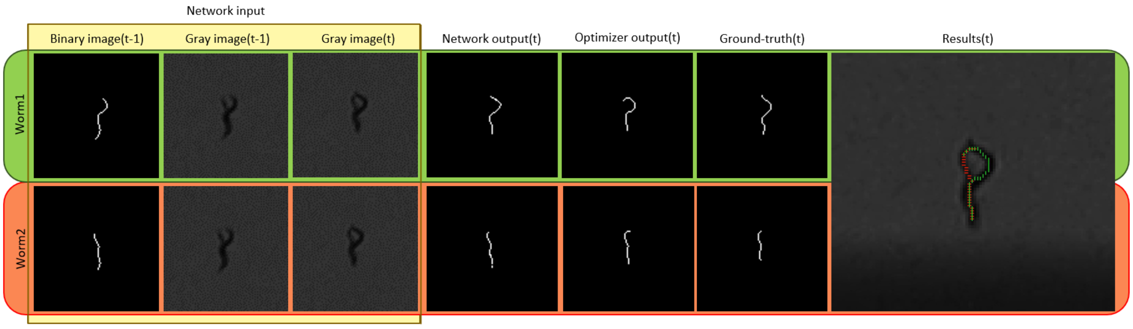


Figure A.32: **Two worms aggregation, example 1.** This image shows the image processing for a two worms aggregation. For this example, while the network output gets a better pose prediction for both worms than the post-processing output (optimizer output), both results are very similar to ground-truth.

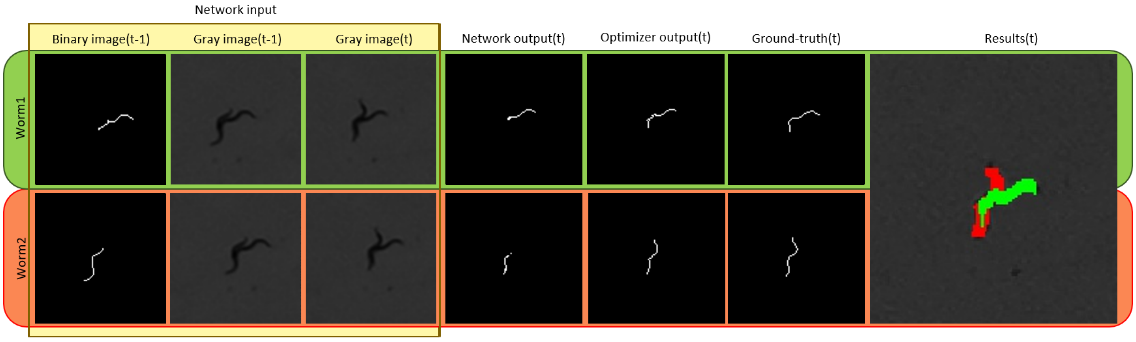


Figure A.33: **Two worms aggregation, example 2.** This image shows the image processing for a two worms aggregation. For this example, the results of the post-processing output (optimizer output) are closer to ground-truth than the network output. The network results are not always the most accurate; some aggregation cases are more complicated than others. But the resulting prediction helps the optimizer to obtain better results.

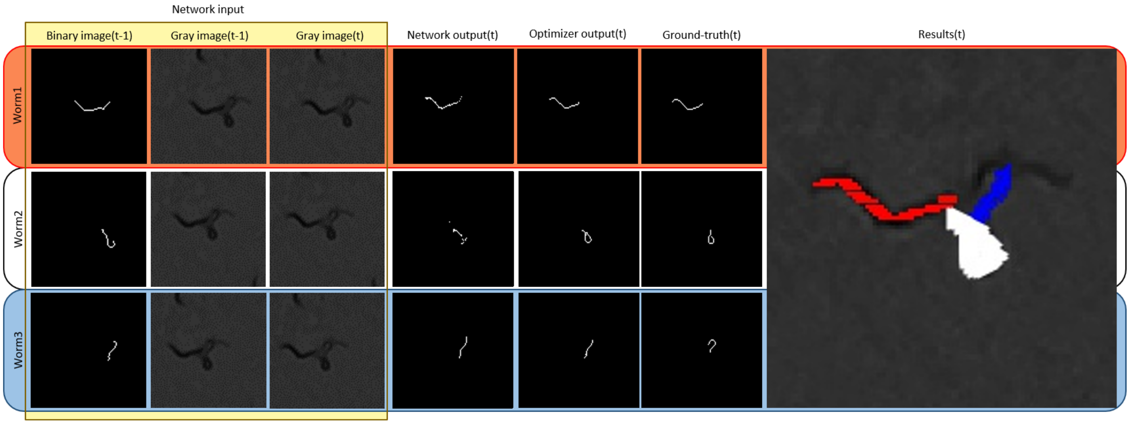


Figure A.34: **Three worms aggregation, example 3.** This image shows the image processing for a three worms aggregation. Although only for worm1 the results are the most accurate, on the other hand, we can see that the network output predicts a non-continuous skeleton, but this helps the optimizer to obtain a better result. For worm2, the output of the network obtains a partial and non-continuous skeleton, this helps the optimizer to obtain a skeleton that although it is not the most accurate, this skeleton allows to preserve the identity of the worm. Finally, worm3 is the same as worm2. This aggregation case is very complicated, but the identity of each individual is preserved and better results are obtained in subsequent images.

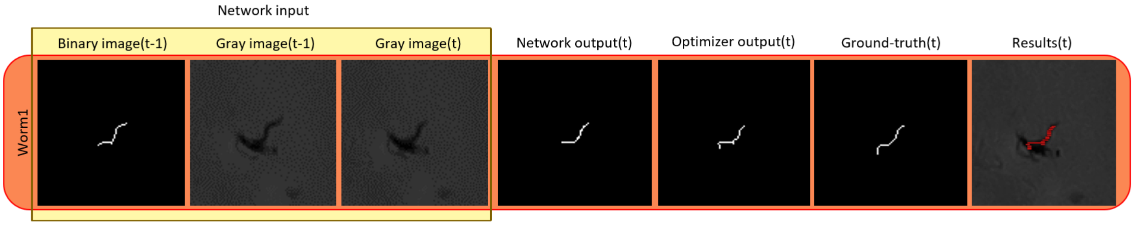


Figure A.35: **Aggregation with noise, example 4.** This image shows the image processing for the aggregation of a worm with noise. For this example, although the results of the post-processing output (optimizer output) are closer to ground-truth than the network output, the results network output helps the optimizer to obtain better results.

Bibliography

[1] M. Alexandre. Pytorch-unet, 2019. Code `https://github.com/milesial/Pytorch-UNet`.

[2] B. Baheti, S. Innani, S. Gajre, and S. Talbar. Eff-unet: A novel architecture for semantic segmentation in unstructured environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1473–1481, Seattle, 2020. IEEE. doi: https://doi.org/10.1109/CVPRW50498.2020.00187.

[3] L. Bargsten and A. Schlaefer. Specklegan: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing. *International journal of computer assisted radiology and surgery*, 15(9): 1427–1436, 2020. doi: https://doi.org/10.1007/s11548-020-02203-1.

[4] K. Bates, K. N. Le, and H. Lu. Deep learning for robust and flexible tracking in behavioral studies for *C. elegans*. *PLOS Computational Biology*, 18(4):e1009942, 2022. doi: https://doi.org/10.1371/journal.pcbi.1009942.

[5] D. Biron and G. Haspel, editors. *C. elegans*. Springer Science+Business Media, New York, 2015. doi: https://doi.org/10.1007/978-1-4939-2842-2.

[6] W. A. Boyd, G. L. Anderson, D. B. Dusenbery, and P. L. Williams. Computer tracking method for assessing behavioral changes in the nematode *Caenorhabditis elegans*. *ASTM SPEC TECH PUBL*, 1381:225–238, 2000. doi: https://doi.org/10.1520/STP14426S.

[7] K. Cao and X. Zhang. An improved res-unet model for tree species classification using airborne high-resolution images. *Remote Sensing*, 12(7), 2020. doi: https://doi.org/10.3390/rs12071128.

[8] L. Chen, M. Strauch, M. Daub, X. Jiang, M. Jansen, H. Luigs, S. Schultz-Kuhlmann, S. Krüssel, and D. Merhof. A cnn framework based on line annotations for detecting nematodes in microscopic images. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 508–512, April 2020. doi: https://doi.org/10.1109/ISBI45749.2020.9098465.

[9] Z. Chen, W. Ouyang, T. Liu, and D. Tao. A shape transformation-based dataset augmentation framework for pedestrian detection. *International Journal of Computer Vision*, 129(4):1121–1138, 2021. doi: https://doi.org/10.1007/s11263-020-01412-0.

[10] Z. Chen, R. Wu, Y. Lin, C. Li, S. Chen, Z. Yuan, S. Chen, and X. Zou. Plant disease recognition model based on improved yolov5. *Agronomy*, 12(2):365, 2022.

[11] P. M. Conn, editor. *Animal Models for the Study of Human Disease*. Sara Tenney, Texas, 2017.

[12] M. De Bono and C. I. Bargmann. Natural variation in a neuropeptide y receptor homolog modifies social behavior and food response in *C. elegans*. *Cell*, 94(5): 679–689, 1998. doi: https://doi.org/10.1016/S0092-8674(00)81609-8.

[13] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo. Yolo v4 for advanced traffic sign recognition with synthetic training data generated by various gan. *IEEE Access*, 9:97228–97242, 2021. doi: https://doi.org/10.1109/ACCESS.2021.3094201.

[14] G. Di Rosa, G. Brunetti, M. Scuto, A. Trovato Salinaro, E. J. Calabrese, R. Crea, C. Schmitz-Linneweber, V. Calabrese, and N. Saul. Healthspan enhancement by olive polyphenols in *C. elegans* wild type and parkinson's models. *International Journal of Molecular Sciences*, 21(11), 2020. doi: https://doi.org/10.3390/ijms21113893.

[15] K. Doshi. Synthetic image augmentation for improved classification using generative adversarial networks. *arXiv preprint arXiv:1907.13576*, 2019. doi: https://doi.org/10.48550/arXiv.1907.13576.

[16] D. B. Dunsenbery. Video camera-computer tracking of nematode *Caenorhabditis elegans* to record behavioral responses. *Journal of chemical ecology*, 11:1239–1247, 1985. doi: https://doi.org/10.1007/BF01024112.

[17] D. B. Dusenbery. Using a microcomputer and video camera to simultaneously track 25 animals. *Computers in Biology and Medicine*, 15(4):169 – 175, 1985. ISSN 0010-4825. doi: https://doi.org/10.1016/0010-4825(85)90058-7.

[18] O. Dzyubachyk, R. Jelier, B. Lehner, W. Niessen, and E. Meijering. Model-based approach for tracking embryogenesis in *Caenorhabditis elegans* fluorescence microscopy data. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5356–5359. IEEE, 2009. doi: https://doi.org/10.1109/IEMBS.2009.5334046.

[19] E. Fontaine, J. Burdick, and A. Barr. Automated tracking of multiple *C. Elegans*. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3716–3719, Aug 2006. doi: https://doi.org/10.1109/IEMBS.2006.260657.

[20] S. Fudickar, E. J. Nustede, E. Dreyer, and J. Bornhorst. Mask r-cnn based *C. Elegans* detection with a diy microscope. *Biosensors*, 11(8):257, 2021. doi: https://doi.org/10.3390/bios11080257.

[21] A. García Garví, J. C. Puchalt, P. E. Layana Castro, F. Navarro Moya, and A.-J. Sánchez-Salmerón. Towards lifespan automation for *Caenorhabditis elegans* based on deep learning: Analysing convolutional and recurrent neural networks for dead or live classification. *Sensors*, 21(14), 2021. doi: https://doi.org/10.3390/s21144943.

[22] T. G.D. and T. N. Nemo: a computational tool for analyzing nematode locomotion. *BMC Neurosci*, 8, 2007. doi: https://doi.org/10.1186/1471-2202-8-86.

[23] J.-H. Hahm, S. Kim, R. DiLoreto, C. Shi, S.-J. V. Lee, C. T. Murphy, and H. G. Nam. *C. elegans* maximum velocity correlates with healthspan and is maintained in worms with an insulin receptor mutation. *Nature communications*, 6(1):1–7, 2015. doi: https://doi.org/10.1038/ncomms9919.

[24] L. Han, P. Tao, and R. R. Martin. Livestock detection in aerial images using a fully convolutional network. *Computational Visual Media*, 5(2):221–228, 2019. doi: https://doi.org/10.1007/s41095-019-0132-5.

[25] L. Hebert, T. Ahamed, A. C. Costa, L. O'Shaugnessy, and G. J. Stephens. Wormpose: Image synthesis and convolutional networks for pose estimation in *C. elegans*. *bioRxiv*, 2020. doi: https://doi.org/10.1101/2020.07.09.193755.

[26] S. Hinterstoisser, O. Pauly, H. Heibel, M. Martina, and M. Bokeloh. An annotation saved is an annotation earned: Using fully synthetic training for object detection. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2787–2796, Seoul, Korea (South), 2019. IEEE. doi: https://doi.org/10.1109/ICCVW.2019.00340.

[27] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059, Barcelona, Spain, 2020. IEEE. doi: https://doi.org/10.1109/ICASSP40776.2020.9053405.

[28] K.-M. Huang, P. Cosman, and W. R. Schafer. Machine vision based detection of omega bends and reversals in *C. elegans*. *Journal of Neuroscience Methods*, 158(2):323 – 336, 2006. ISSN 0165-0270. doi: https://doi.org/10.1016/j.jneumeth.2006.06.007.

[29] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/ioffe15.html`.

[30] H. Iqbal. Harisiqbal88/plotneuralnet v1.0.0, 2018. Code `https://github.com/HarisIqbal88/PlotNeuralNet`.

[31] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021. doi: https://doi.org/10.1038/s41592-020-01008-z.

[32] S. Jaensch, M. Decker, A. A. Hyman, and E. W. Myers. Automated tracking and analysis of centrosomes in early *Caenorhabditis elegans* embryos. *Bioinformatics*, 26 (12):i13–i20, 2010. doi: https://doi.org/10.1093/bioinformatics/btq190.

[33] A. Javer, M. Currie, C. W. Lee, J. Hokanson, K. Li, C. N. Martineau, E. Yemini, L. J. Grundy, C. Li, Q. Ch'ng, W. R. Schafer, E. A. A. Nollen, R. Kerr, and A. E. X. Brown. An open-source platform for analyzing and sharing worm-behavior data. *Nature Methods*, 15(9):645–646, Sept. 2018. ISSN 1548-7105. doi: https://doi.org/10.1038/s41592-018-0112-1.

[34] A. Javer, A. E. Brown, I. Kokkinos, and J. Rittscher. Identification of *C. elegans* strains using a fully convolutional neural network on behavioural dynamics. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, volume 11134, pages 0–0. Springer, Cham, 2019. doi: https://doi.org/10.1007/978-3-030-11024-6_35.

[35] S.-K. Jung, B. Aleman-Meza, C. Riepe, and W. Zhong. Quantworm: a comprehensive software package for *Caenorhabditis elegans* phenotypic assays. *PLoS One*, 9(1):e84830, 2014. doi: https://doi.org/10.1371/journal.pone.0084830.

[36] M. Kiel, D. Berh, J. Daniel, N. Otto, A. t. Steege, X. Jiang, E. Liebau, and B. Risse. A multi-purpose worm tracker based on fim, 2018.

[37] M. Kim, D. Knoefler, E. Quarles, U. Jakob, and D. Bazopoulou. Automated phenotyping and lifespan assessment of a *C. elegans* model of parkinson's disease. *Translational Medicine of Aging*, 4:38 – 44, 2020. ISSN 2468-5011. doi: https://doi.org/10.1016/j.tma.2020.04.001.

[38] M. Koopman, Q. Peter, R. I. Seinstra, M. Perni, M. Vendruscolo, C. M. Dobson, T. P. Knowles, and E. A. Nollen. Assessing motor-related phenotypes of *Caenorhabditis elegans* with the wide field-of-view nematode tracking platform. *Nature protocols*, 15(6):2071–2106, 2020. doi: https://doi.org/10.1038/s41596-020-0321-9.

[39] A. Koul, S. Ganju, and M. Kasam. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*, pages 679–680. O'Reilly Media, 2019. ISBN 9781492034834. URL `https://books.google.es/books?id=GcS2DwAAQBAJ`.

[40] S. Kumar, B. M. Egan, Z. Kocsisova, D. L. Schneider, J. T. Murphy, A. Diwan, and K. Kornfeld. Lifespan extension in *C. elegans* caused by bacterial colonization of the intestine and subsequent activation of an innate immune response. *Developmental Cell*, 49(1):100–117.e6, 2019. ISSN 1534-5807. doi: https://doi.org/10.1016/j.devcel.2019.03.010.

[41] P. E. Layana Castro, J. C. Puchalt, and A.-J. Sánchez-Salmerón. Improving skeleton algorithm for helping *Caenorhabditis elegans* trackers. *Scientific Reports*, 10(1):22247, 2020. doi: https://doi.org/10.1038/s41598-020-79430-8.

[42] P. E. Layana Castro, J. C. Puchalt, A. García Garví, and A.-J. Sánchez-Salmerón. *Caenorhabditis elegans* multi-tracker based on a modified skeleton algorithm. *Sensors*, 21(16), 2021. ISSN 1424-8220. doi: https://doi.org/10.3390/s21165622.

[43] P. E. Layana Castro, G. Garví, N. M. Antonio, Francisco, and A.-J. Sánchez-Salmerón. Skeletonizing *Caenorhabditis elegans* based on u-net architectures trained with a multi-worm low-resolution synthetic dataset. *International Journal of Computer Vision*, 2023. doi: https://doi.org/10.1007/s11263-023-01818-6.

[44] K. N. Le, M. Zhan, Y. Cho, J. Wan, D. S. Patel, and H. Lu. An automated platform to monitor long-term behavior and healthspan in *Caenorhabditis elegans* under precise environmental control. *Communications biology*, 3(1):1–13, 2020. doi: https://doi.org/10.1038/s42003-020-1013-2.

[45] N. Leonard and A. G. Vidal-Gadea. Affordable *Caenorhabditis elegans* tracking system for classroom use. *microPublication biology*, 2021, 2021. doi: https://doi.org/10.17912/micropub.biology.000377.

[46] H. Li, J. Fang, S. Liu, X. Liang, X. Yang, Z. Mai, M. T. Van, T. Wang, Z. Chen, and D. Ni. Cr-unet: A composite network for ovary and follicle segmentation in ultrasound images. *IEEE Journal of Biomedical and Health Informatics*, 24(4):974–983, 2020. doi: https://doi.org/10.1109/JBHI.2019.2946092.

[47] S. Li, S. Günel, M. Ostrek, P. Ramdya, P. Fua, and H. Rhodin. Deformation-aware unpaired image translation for pose estimation on laboratory animals. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13155–13165, June 2020. doi: https://doi.org/10.1109/CVPR42600.2020.01317.

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. doi: https://doi.org/10.1007/978-3-319-10602-1_48.

[49] X. Liu, T. Zhou, M. Lu, Y. Yang, Q. He, and J. Luo. Deep learning for ultrasound localization microscopy. *IEEE Transactions on Medical Imaging*, 39(10):3064–3078, 2020. doi: https://doi.org/10.1109/TMI.2020.2986781.

[50] T. Lorimer, R. Goodridge, A. K. Bock, V. Agarwal, E. Saberski, G. Sugihara, and S. A. Rifkin. Tracking changes in behavioural dynamics using prediction error. *Plos one*, 16(5):e0251053, 2021. doi: https://doi.org/10.1371/journal.pone.0251053.

[51] L. Mais, P. Hirsch, and D. Kainmueller. Patchperpix for instance segmentation. In *European Conference on Computer Vision*, volume 12370, pages 288–304. Springer, Springer, Cham, 2020. doi: https://doi.org/10.1007/978-3-030-58595-2_18.

[52] M. R. Mane, A. A. Deshmukh, and A. J. Iliff. Head and tail localization of *C. elegans*. *arXiv preprint arXiv:2001.03981*, 2020. doi: https://doi.org/10.48550/arXiv.2001.03981.

[53] C. Mayershofer, T. Ge, and J. Fottner. Towards fully-synthetic training for industrial applications. In *LISS 2020*, pages 765–782. Springer, Singapore, 2021. doi: https://doi.org/10.1007/978-981-33-4359-7_53.

[54] J. E. McManigle, R. R. Bartz, and L. Carin. Y-net for chest x-ray preprocessing: Simultaneous classification of geometry and segmentation of annotations. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 1266–1269, Montreal, QC, Canada, 2020. IEEE. doi: https://doi.org/10.1109/EMBC44109.2020.9176334.

[55] S. Moradi, M. G. Oghli, A. Alizadehasl, I. Shiri, N. Oveisi, M. Oveisi, M. Maleki, and J. Dhooge. Mfp-unet: A novel deep learning based approach for left ventricle segmentation in echocardiography. *Physica Medica*, 67:58–69, 2019. ISSN 1120-1797. doi: https://doi.org/10.1016/j.ejmp.2019.10.001.

[56] S. Nagy, M. Goessling, Y. Amit, and D. Biron. A generative statistical algorithm for automatic detection of complex postures. *PLOS Computational Biology*, 11(10):1–23, 10 2015. doi: https://doi.org/10.1371/journal.pcbi.1004517.

[57] A. Olsen and M. S. Gill, editors. *Ageing: Lessons from C. elegans*. Springer International Publishing, 2017. doi: https://doi.org/10.1007/978-3-319-44703-2.

[58] A. Olsen and M. S. Gill, editors. *Ageing: Lessons from C. elegans*. Springer International Publishing, Switzerland, 2017. doi: https://doi.org/10.1007/978-3-319-44703-2.

[59] C. Padubidri, A. Kamilaris, S. Karatsiolis, and J. Kamminga. Counting sea lions and elephants from aerial photography using deep learning with density maps. *Animal Biotelemetry*, 9(1):1–10, 2021. doi: https://doi.org/10.1186/s40317-021-00247-x.

[60] A. Pashevich, R. Strudel, I. Kalevatykh, I. Laptev, and C. Schmid. Learning to augment synthetic images for sim2real policy transfer. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2651–2657, Macau, China, 2019. IEEE. doi: https://doi.org/10.1109/IROS40897.2019.8967622.

[61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

[62] J. N. Pitt, N. L. Strait, E. M. Vayndorf, B. W. Blue, C. H. Tran, B. E. Davis, K. Huang, B. J. Johnson, K. M. Lim, S. Liu, et al. Wormbot, an open-source robotics platform for survival and behavior analysis in *C. elegans*. *GeroScience*, 41(6):961–973, 2019. doi: https://doi.org/10.1007/s11357-019-00124-9.

[63] E. Plebani, N. P. Biscola, L. A. Havton, B. Rajwa, A. S. Shemonti, D. Jaffey, T. Powley, J. R. Keast, K.-H. Lu, and M. M. Dundar. High-throughput segmentation of unmyelinated axons by deep learning. *Scientific Reports*, 12(1):1–16, 2022. doi: https://doi.org/10.1038/s41598-022-04854-3.

[64] J. C. Puchalt, A.-J. Sánchez-Salmerón, P. Martorell Guerola, and S. Genovés Martínez. Active backlight for automating visual monitoring: An analysis of a lighting control technique for *Caenorhabditis elegans* cultured on standard petri plates. *PLOS ONE*, 14(4):1–18, 04 2019. doi: 10.1371/journal.pone.0215548. URL `https://doi.org/10.1371/journal.pone.0215548`.

[65] J. C. Puchalt, P. E. Layana Castro, and A.-J. Sánchez-Salmerón. Reducing results variance in lifespan machines: an analysis of the influence of vibrotaxis on wild-type *Caenorhabditis elegans* for the death criterion. *Sensors*, 20(21):5981, 2020. doi: https://doi.org/10.3390/s20215981.

[66] J. C. Puchalt, A.-J. Sánchez-Salmerón, E. Ivorra, S. Genovés Martínez, R. Martínez, and P. Martorell Guerola. Improving lifespan automation for *Caenorhabditis elegans* by using image processing and a post-processing adaptive data filter. *Scientific reports*, 10(1):8729, May 2020. ISSN 2045-2322. doi: https://doi.org/10.1038/s41598-020-65619-4.

[67] J. C. Puchalt, A.-J. Sánchez-Salmerón, E. Ivorra, S. Llopis, R. Martínez, and P. Martorell. Small flexible automated system for monitoring *Caenorhabditis elegans* lifespan based on active vision and image processing techniques. *Scientific Reports*, 11(1): 12289, 2021. doi: https://doi.org/10.1038/s41598-021-91898-6.

[68] J. C. Puchalt, J. F. Gonzalez-Rojo, A. P. Gómez-Escribano, R. P. Vázquez-Manrique, and A.-J. Sánchez-Salmerón. Multiview motion tracking based on a cartesian robot to monitor *Caenorhabditis elegans* in standard petri dishes. *Scientific reports*, 12(1): 1–11, 2022. doi: https://doi.org/10.1038/s41598-022-05823-6.

[69] S. Qamar, H. Jin, R. Zheng, P. Ahmad, and M. Usama. A variant form of 3d-unet for infant brain segmentation. *Future Generation Computer Systems*, 108:613–623, 2020. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2019.11.021.

[70] D. Ramot, B. E. Johnson, T. L. Berry, Jr, L. Carnell, and M. B. Goodman. The parallel worm tracker: A platform for measuring average speed and drug-induced paralysis in nematodes. *PLOS ONE*, 3(5):1–7, 05 2008. doi: https://doi.org/10.1371/journal.pone.0002208.

[71] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. doi: https://doi.org/10.1109/CVPR.2016.91.

[72] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. doi: https://doi.org/10.1109/TPAMI.2016.2577031.

[73] C. Restif, C. Ibañez-Ventoso, M. Driscoll, and D. Metaxas. Tracking *C. elegans* swimming for high-throughput phenotyping. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1542–1548, 2011. doi: https://doi.org/10.1109/ISBI.2011.5872695.

[74] C. Restif, C. Ibáñez-Ventoso, M. M. Vora, S. Guo, D. Metaxas, and M. Driscoll. Celest: Computer vision software for quantitative analysis of *C. elegans* swim behavior reveals novel features of locomotion. *PLOS Computational Biology*, 10(7):1–12, 07 2014. doi: https://doi.org/10.1371/journal.pcbi.1003702.

[75] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, June 2019. doi: https://doi.org/10.1109/CVPR.2019.00075.

[76] N. B. Rizvandi, A. Pizurica, and W. Philips. Machine vision detection of isolated and overlapped nematode worms using skeleton analysis. In *2008 15th IEEE International Conference on Image Processing*, pages 2972–2975, Oct 2008. doi: https://doi.org/10.1109/ICIP.2008.4712419.

[77] N. B. Rizvandi, A. Pižurica, F. Rooms, and W. Philips. Skeleton analysis of population images for detection of isolated and overlapped nematode *C. elegans*. In *2008 16th European Signal Processing Conference*, pages 1–5, Aug 2008.

[78] C. Rogers, A. Persson, B. Cheung, and M. de Bono. Behavioral motifs and neural pathways coordinating o2 responses and aggregation in *C. elegans*. *Current biology*, 16(7):649–659, 2006. doi: https://doi.org/10.1016/j.cub.2006.03.023.

[79] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, volume 9351, pages 234–241. Springer, Springer, Cham, 2015. doi: https://doi.org/10.1007/978-3-319-24574-4_28.

[80] N. Roussel, C. A. Morton, F. P. Finger, and B. Roysam. A computational model for *C. elegans* locomotory behavior: Application to multiworm tracking. *IEEE Transactions on Biomedical Engineering*, 54(10):1786–1797, Oct 2007. ISSN 1558-2531. doi: https://doi.org/10.1109/TBME.2007.894981.

[81] J. C. Russ and F. B. Neal. *The Image Processing Handbook*, pages 479–480. CRC Press, Inc., USA, 7th edition, 2015. ISBN 149874026X.

[82] D. Schraml. Physically based synthetic image generation for machine learning: a review of pertinent literature. In *Photonics and Education in Measurement Science 2019*, volume 11144, page 111440J, Jena, Germany, 2019. International Society for Optics and Photonics. doi: https://doi.org/10.1117/12.2533485.

[83] S. H. Simonetta and D. A. Golombek. An automated tracking system for *Caenorhabditis elegans* locomotor behavior and circadian studies application. *Journal of neuroscience methods*, 161(2):273–280, 2007. doi: https://doi.org/10.1016/j.jneumeth.2006.11.015.

[84] M. B. Sokolowski. Social interactions in “simple” model systems. *Neuron*, 65(6): 780–794, 2010. doi: https://doi.org/10.1016/j.neuron.2010.03.007.

[85] J. H. Sossa. An improved parallel algorithm for thinning digital patterns. *Pattern Recognition Letters*, 10(2):77–80, 1989. doi: https://doi.org/10.1016/0167-8655(89)90070-6.

[86] M. Spensley, S. Del Borrello, D. Pajkic, and A. G. Fraser. Acute effects of drugs on *Caenorhabditis elegans* movement reveal complex responses and plasticity. *G3: Genes, Genomes, Genetics*, 8(9):2941–2952, 2018. doi: https://doi.org/10.1534/g3.118.200374.

[87] J. Srinivasan, S. H. von Reuss, N. Bose, A. Zaslaver, P. Mahanti, M. C. Ho, O. G. O’Doherty, A. S. Edison, P. W. Sternberg, and F. C. Schroeder. A modular library of small molecule signals regulates social behaviors in *Caenorhabditis elegans*. *PLoS biology*, 10(1):e1001237, 2012. doi: https://doi.org/10.1371/journal.pbio.1001237.

[88] T. Stiernagle. Maintenance of *C. elegans*, 2006.

[89] N. A. Swierczek, A. C. Giles, C. H. Rankin, and R. A. Kerr. High-throughput behavioral analysis in *C. elegans*. *Nature Methods*, 8(7):592–598, July 2011. ISSN 1548-7105. doi: https://doi.org/10.1038/nmeth.1625.

[90] R. Sznitman, M. Gupta, G. D. Hager, P. E. Arratia, and J. Sznitman. Multi-environment model estimation for motility analysis of *Caenorhabditis elegans*. *PLOS ONE*, 5(7), 07 2010. doi: https://doi.org/10.1371/journal.pone.0011631.

[91] P. Tang, Q. Liang, X. Yan, S. Xiang, W. Sun, D. Zhang, and G. Coppola. Efficient skin lesion segmentation using separable-unet with stochastic weight averaging. *Computer Methods and Programs in Biomedicine*, 178:289–301, 2019. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2019.07.005.

[92] E. Teo, S. Y. J. Lim, S. Fong, A. Larbi, G. D. Wright, N. Tolwinski, and J. Gruber. A high throughput drug screening paradigm using transgenic *Caenorhabditis elegans* model of alzheimer’s disease. *Translational Medicine of Aging*, 4:11 – 21, 2020. ISSN 2468-5011. doi: https://doi.org/10.1016/j.tma.2019.12.002.

[93] K. Trebing, T. Stanczyk, and S. Mehrkanoon. Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. *Pattern Recognition Letters*, 145:178–186, 2021. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2021.01.036.

[94] P. Tschandl, C. Sinz, and H. Kittler. Domain-specific classification-pretrained fully convolutional network encoders for skin lesion segmentation. *Computers in Biology and Medicine*, 104:111–116, 2019. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2018.11.010.

[95] V. Uhlmann and M. Unser. Tip-seeking active contours for bioimage segmentation. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 544–547, 2015. doi: https://doi.org/10.1109/ISBI.2015.7163931.

[96] D. Wang, Z. Lu, and Z. Bao. Augmenting *C. elegans* microscopic dataset for accelerated pattern recognition. *arXiv preprint arXiv:1906.00078*, 2019. doi: https://doi.org/10.48550/arXiv.1906.00078.

[97] L. Wang, S. Kong, Z. Pincus, and C. Fowlkes. Celeganser: Automated analysis of nematode morphology and age. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4164–4173, Seattle, WA, USA, 2020. IEEE. doi: https://doi.org/10.1109/CVPRW50498.2020.00492.

[98] S. Wiehman and H. de Villiers. Semantic segmentation of bioimages using convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 624–631, Vancouver, BC, Canada, 2016. IEEE. doi: https://doi.org/10.1109/IJCNN.2016.7727258.

[99] O. Wiles and A. Zisserman. Learning to predict 3d surfaces of sculptures from single and multiple views. *International Journal of Computer Vision*, 127(11):1780–1800, 2019. doi: https://doi.org/10.1007/s11263-018-1124-0.

[100] P. B. Winter, R. M. Brielmann, N. P. Timkovich, H. T. Navarro, A. Teixeira-Castro, R. I. Morimoto, and L. A. N. Amaral. A network approach to discerning the identities of *C. elegans* in a free moving population. *Scientific Reports*, 6:34859, Oct. 2016. doi: https://doi.org/10.1038/srep34859.

[101] C. Wählby, L. Kamentsky, Z. Liu, T. Riklin Raviv, A. Conery, E. O'Rourke, K. Sokolnicki, O. Visvikis, V. Ljosa, J. Irazoqui, P. Golland, G. Ruvkun, F. Ausubel, and A. Carpenter. An image analysis toolbox for high-throughput *C. elegans* assays. *Nature methods*, 9:714–6, 04 2012. doi: https://doi.org/10.1038/nmeth.1984.

[102] C.-C. J. Yu, D. M. Raizen, and C. Fang-Yen. Multi-well imaging of development and behavior in *Caenorhabditis elegans*. *Journal of Neuroscience Methods*, 223:35–39, 2014. ISSN 0165-0270. doi: https://doi.org/10.1016/j.jneumeth.2013.11.026.

[103] X. Yu, M. S. Creamer, F. Randi, A. K. Sharma, S. W. Linderman, and A. M. Leifer. Fast deep neural correspondence for tracking and identifying neurons in *C. elegans* using semi-synthetic training. *eLife*, 10:e66410, jul 2021. ISSN 2050-084X. doi: https://doi.org/10.7554/eLife.66410.

[104] X. Zhao, Y. Yuan, M. Song, Y. Ding, F. Lin, D. Liang, and D. Zhang. Use of unmanned aerial vehicle imagery and deep learning unet to extract rice lodging. *Sensors*, 19 (18), 2019. doi: https://doi.org/10.3390/s19183859.