



SegElegans: Instance segmentation using dual convolutional recurrent neural network decoder in *Caenorhabditis elegans* microscopic images

Pablo E. Layana Castro ^a, Konstantinos Kounakis ^{b,c} , Antonio García Garvía ^a , Ilias Gkikas ^{c,d} ,
Ioannis Tsiamantas ^{c,d} , Nektarios Tavernarakis ^{b,c,*}, Antonio-José Sánchez-Salmerón ^a ,*

^a Universitat Politècnica de València, Instituto de Automática e Informática Industrial, Camino de Vera S/n, Edificio 8G Acceso D, Valencia, 46022, Valencia, Spain

^b Department of Basic Sciences, Faculty of Medicine, University of Crete, Heraklion, 71110, Crete, Greece

^c Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas, Heraklion, 71110, Crete, Greece

^d Department of Biology, School of Sciences and Engineering, University of Crete, Heraklion, 71110, Crete, Greece

ARTICLE INFO

Keywords:

Caenorhabditis elegans
Neural network
Segmentation
Skeletonizing
Microscopic images
Automated analysis

ABSTRACT

Caenorhabditis elegans is a great model for exploring organismal, cellular, and subcellular biology through optical and fluorescence microscopy, with its research applications steadily expanding. However, manual processing of numerous microscopic images is prone to errors and demands significant labor due to worms tendency to touch or cluster with each other. Here, we present a new system for segmenting whole-body instances of *Caenorhabditis elegans* in microscopic images (referred to as SegElegans), employing a combination of neural network architecture and conventional image processing techniques. Our method effectively overcomes previous challenges and resolves many instances of contact and overlap between worms in highly populated images in a timely manner. The results obtained show an average Intersection over Union value of 96.3% per worm and an average improvement of 6% over other existing methods for automated analysis of worm images. SegElegans is a user-friendly application for *Caenorhabditis elegans* segmentation that will benefit whole-worm phenotypic screenings essential for studying development, behavior, aging, and disease.

1. Introduction

Having been used as a laboratory organism for more than half a century, the nematode *Caenorhabditis elegans* (*C. elegans*) has been instrumental in providing knowledge nearly in all aspects of biology [1]. Its short lifespan of about three weeks, its compact size of 1 mm, its transparency, and its high amount of genomic conservation relative to the human genome, make this animal model very attractive to study neurodegenerative diseases, and create/test new drugs [2]. Worms can be grown abundantly in small spaces, such as Petri dishes, facilitating cost-effective and scalable assays. Currently, applications continue to be developed taking advantage of all these characteristics to test different treatments and medicines in the fields of biology, pharmaceuticals, and cosmetics.

For phenotypic screening assays, researchers usually perform manual labeling tasks to segment/label each of the worms one by one and thus be able to obtain information about each of them [3–5]. The process can be quite tedious and time-consuming, especially when trying

to gather data from large amounts of worms to achieve high-quality statistics. A basic whole worm selection (closely surrounding the whole body) by an experienced user can take about 20 s per worm, while a high precision selection (as close to the actual edge of the animals as image resolution allows), which is often needed in images with high numbers of touching and overlapping worms, can take a full minute per worm. The produced full-body high-precision masks/segmentations allow for quick and easy morphological measurements (body dimensions) in brightfield images. They can also be transferred to darkfield fluorescence images that have been taken simultaneously by the same system for more “advanced” types of analyses, like quantification of fluorescence intensity per worm [6,7], use of specialized algorithms to measure fluorescent marked particle amount and dimensions [8], measure protein aggregation [9,10], or assess organelle network structure. Obtaining individual information and using it for statistical analysis of the population is the best option for scientific analysis. This produces

* Corresponding authors.

E-mail addresses: pablacas@doctor.upv.es (P.E.L. Castro), kostas.kounakis@imbb.forth.gr (K. Kounakis), angar25a@upv.edu.es (A.G. Garvía), igkikas@imbb.forth.gr (I. Gkikas), ioan.tsiamantas@gmail.com (I. Tsiamantas), tavernarakis@imbb.forth.gr (N. Tavernarakis), asanchez@isa.upv.es (A.-J. Sánchez-Salmerón).

<https://doi.org/10.1016/j.complbiomed.2025.110012>

Received 3 September 2024; Received in revised form 4 March 2025; Accepted 7 March 2025

0010-4825/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

higher-quality statistics than alternative options (like taking whole image mean values and relying on just summary statistics) by allowing us to properly quantify population variance; this can be especially helpful at revealing sub-populations with distinct behavior that has potential biological significance. For instance, there is a biologically relevant difference between a treatment reducing a protein level by 50% in all animals or reducing it by 90% in half of the animals and per-worm analysis is needed to detect it. Thus, when trying to automate these experiments, full masks/segmentations of each individual worm are preferred.

Currently, some applications automatically segment worm populations into individual masks using image processing. This can be a challenge, due to the large number of pixels in the image, and because algorithms may not be strong enough to distinguish highly complex features. It can be even more difficult when there are areas/segmentations of interest that are connected/overlapping or when the images contain noise that can lead to segmentations that do not correspond to true areas of interest. In images of *C. elegans*, noise typically comes in the form of eggs and small larvae, organic waste, or other foreign objects of no interest. Their intensity in gray levels is similar, making it difficult to differentiate segmentations of worms from noise, and the issue is even worse when this noise is in contact or close to real worms. Nevertheless, traditional image processing applications have demonstrated some good results of individual worm masks/segmentations on population images by using skeleton segmentation, analyzing poses, intersections, ends [11–13], or active contours [14]. But detecting and segmenting pixels or areas of interest with traditional image processing techniques requires some advanced operations and designing robust algorithms capable of identifying patterns, extracting relevant features or filtering noise, not to mention of dealing with cases of connected, overlapping segmentations. All of this leads to increased code complexity, longer processing times and increased computational load. On the other hand, deep-learning techniques and neural networks for image segmentation have demonstrated more reliable results, being faster and more robust in the presence of noise [15–18], even using DIY Microscopes [19]. However, training these methods requires a big and varied dataset. The cost of a labeled dataset is also high, but the results that these techniques offer are worthwhile compared to manually labeling hundreds or thousands of experiments to be performed. The currently best-known neural network architecture to obtain segmentations is U-Net [20]. This type of architecture was first designed to segment biomedical images, and nowadays researchers configure and adapt this network architecture to obtain better results with different datasets [21]; with biological images [22], objects [23], *C. elegans* [24], etc.

Although there are other architectures for pixel segmentation, such as Mask R-CNN [25], which has a robust structure capable of detecting objects and offering high segmentation accuracy, it requires more computational resources. As a result, its training and parameter tuning tend to be slower compared to lighter models, such as U-Net [20]. On the other hand, models based on U-Net architectures, such as EmbedSeg [26], or variations of U-Net [27–29], are simpler and easier to implement, not to mention that their accuracy is just as high and memory usage is low, which makes them ideal for complex tasks and large datasets.

In our previous approach [24], we solved multiple worm overlaps for low-resolution images. This method used 2 images (previous and present) and a neural network to predict the actual pose (skeleton) of each worm respectively. Although very good results were obtained, there were cases that were difficult to solve, even visually. Cases such as worms in contact, in parallel, partially in parallel, etc. Visually they may appear to be touching, but it may be the case that they are overlapping. The problem with low-resolution is the poor grey-level information. You cannot distinguish worm parts such as edges, overlapping parts, and non-overlapping parts, because the grey-level is the same. On the other hand, in high-resolution microscopic images, these parts have different grey-levels and can be segmented using

advanced image processing techniques such as convolutional neural networks. A correct segmentation of these parts can help to correctly identify each worm in densely populated images.

This work presents a new application to obtain individual whole-body segmentation of *C. elegans* in microscopic images referred to as SegElegans. Our approach utilizes a hybrid methodology that combines U-Net architecture with double decoding with traditional image processing techniques. SegElegans shows remarkable performance in obtaining individual masks/segmentations of worms in densely populated images and can resolve many of the cases of contact and overlapping between worms and with image noise with great effectiveness.

2. Methods

2.1. *Caenorhabditis elegans* strain and culture conditions

All image training has been done with WT worms (N2 Bristol) grown as standard [30] on Nematode Growth Medium (NGM) covered with *E. coli* (strains OP50 or HT115). Animals were grown at 20° for a varied number of days in order to achieve higher variation in animal ages and sizes for the training/validation/test datasets. The range of *C. elegans* populations used in each image were between 3 to 30 worms. This variation in population density allowed us to train models capable of accurately obtaining and distinguishing segmentation regions of interest.

2.2. Image capture system

Image acquisition for the training/validation/test dataset was performed using the EVOS FL Auto 2 epifluorescent microscope. EVOS, it is a ThermoFisher Scientific product (HQ in Waltham Massachusetts) and is made by Life Technologies Corporation in Bothell Washington (the western Washington). The captured images had various cases of overlap and contact between *C. elegans*, with worm populations varying between 3 and 30 worms. For greater variability of the dataset, different exposure/brightness/contrast settings were performed during image acquisition, and rotation and flip operations were also introduced during training/validation. These configurations allowed us to develop more robust deep-learning models in the presence of noise and various problematic segmentation scenarios. As a result, better segmentations of individual *C. elegans* were obtained, resolving many of the cases of overlap and contact between them.

2.3. Proposed neural network

This paper aims to obtain segmentations or individual masks in microscopic gray images of *C. elegans*. For this, we use two segmentation images. The first image (Fig. 1 Output1) contains segmentations of: background (black segmentation), worm edges (red segmentation), overlapping parts between worms (green segmentation), and non-overlapping worm parts (blue segmentation). The second image (Fig. 1 Output2) contains background (black segmentation) and worm skeleton (white segmentation).

The proposed neural network model (Fig. 1) consists of three neural network blocks, one encoder block (orange and yellow blocks) and two decoder blocks (purple and light-blue blocks). The encoder block belongs to the SmaAt AT neural network model [21], this model is based on the U-Net architecture [20] with the difference that instead of using regular convolutional operations it uses depth-separable convolutions and adds CBAM attention mechanism to the encoder. CBAM blocks are applied after double convolution blocks (DoubleConv2D) and allow important features to be identified. For the decoder block, we replace the double convolution typical of U-Net architectures with convolution blocks and Long Short-Term Memory block (ConvLSTM) [31]. Although the ConvLSTM block is designed to work with temporal sequences, in this case the dimension of the time axis, T , is set to 1, which means

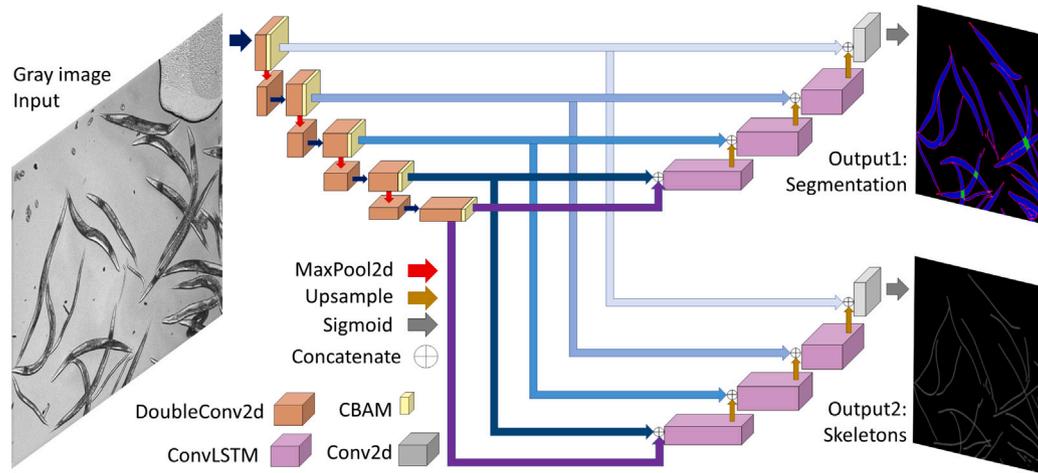


Fig. 1. Proposed neural network model. The proposed model uses the encoder of the SmaAt AT model proposed by [21] (orange and yellow blocks) and for the decoder part uses the ConvLSTM block [31] (purple blocks) instead of the typical double convolution of U-Net architectures.

that the model perceives a single sequence for each input. This allowed the model to learn complex spatial relationships between concatenated features. This convLSTM block [31] according to the author is a little different from the original article [32]. This convLSTM block simplifies the original block by removing direct connections to the previous cell state and using an intermediate state gt , which makes the cell update more modular. These differences make the model easier to implement and less computationally expensive.

2.4. Training/validation and testing method

The dataset consists of 100 images of size 1328×1048 , 50 images for training/validation (70% training, 30% validation), and 50 images for testing. Due to the dimensions of the input image and hardware limitations we divide the input image and the ground-truth images into 512×512 pixel windows. As a data augmentation technique and for greater variability in training/validation, each image was divided into 64 parts using 512×512 pixel windows. Each crop was performed in steps of 102 pixels for the X-axis (1328px dimension), and steps of 67 pixels for the Y-axis (1048px dimension). Rotations of 90, 180, 270, and 360 degrees and horizontal and vertical flips were also applied. Before starting the training and validation, the name of each image and the position of the 64 512×512 windows were saved in two lists. During training and validation, rotations and flips were applied randomly.

The loss function used to train/validate the networks were the PyTorch function “BCEWithLogitsLoss()” and “CrossEntropyLoss()”. CrossEntropyLoss() is a Pytorch loss function for class classification. This function was used to train/validate output1 (Fig. 1), which contains 4 classes: Background segmentation (black), edge segmentation (red), overlapping worm segmentation (green), and non-overlapping worm segmentation (blue). On the other hand, Binary Cross Entropy Loss with Logits Loss or BCEWithLogitsLoss() is another Pytorch loss function for binary classification, that is, for single-unit output classification. This function was used to train/validate Output2 (worm skeleton segmentation). A learning rate scheduling function was used for the training phase. This function allowed the learning rate to be lowered progressively by multiplying the initial learning rate by a given function (Eq. (1)).

$$\lambda_{fn} = \left(1 - \frac{\text{epoch}}{\text{total_epoch}}\right)^{0.9} \quad (1)$$

Where epoch is the current epoch number ranging from 0 to 499, and total_epoch is the total epoch value of the training(500). The learning rate value was 0.0001, so for epoch 0 the learning rate value was 0.0001, for epoch 2 it was $9.9628\text{E}-5$, for epoch 100 it was $7.6508\text{E}-5$ and for the final epoch (499) it was $1.8002\text{E}-7$.

2.5. Post-processing method

The proposed neural network model has two images as output. To obtain the final segmentations that we will use in the post-processing method (Fig. 2b and Fig. 2c), a threshold of 0.5 was first applied to each output, i.e., if the pixel value is greater than or equal to 0.5 its value is one otherwise its value is zero. The first image after applying the threshold of 0.5 contains the segmentations of worm edge parts, overlapping parts, and non-overlapping worm parts. The second image contains the skeleton segmentation of all the worms in the image. These two images were used to design a post-processing method capable of separating overlapping worms or those in contact with each other (Fig. 2f), i.e., obtaining individual segmentations or masks of each worm in an input gray image (Fig. 2a).

The post-processing method consists of 2 parts, the first part is responsible for obtaining the segmentations of non-overlapping worms (Fig. 2d), i.e., individual worms, worms that have contact at the edges or at the ends. To obtain this segmentation we analyzed the overlaps in these areas (Fig. 3 a, b, c) using the skeleton segmentation image (Fig. 3 d,e,f) to identify true-overlaps from false-overlaps segmentations. The criterion is that if those overlapping worm parts (green color) contain skeleton segmentation in the same position, then it belongs to true-overlap segmentations, otherwise it belongs to false-overlap. The false-overlap result and edge segmentation (red channel Fig. 2b) were used to separate non-overlapping worms (Fig. 3d) from overlapping worms (Fig. 3e). Edge and skeleton segmentation were only used in overlap or contact cases as a preventive step to filter out false positives.

The second part of the post-processing is responsible for obtaining the worm segmentations in true-overlaps (Fig. 2e), for this it divides the segmentations into none-overlapping worm parts (color segmentations) and overlapping parts (green segmentation) using the edge segmentation (red channel Fig. 2b). A recursive function is used to join the parts and create a complete worm segmentation (Fig. 4b). This function takes a none-overlapping part, e.g. part 1 (light-green part Fig. 4a), and finds the next best none-overlapping part (part 4) that is connected to the same overlapping part, this is repeated until there are no more parts to join. The next best none-overlapping part was obtained by analyzing the angles of the other none-overlapping parts connected to the same overlapping part. If the angle was equal or with a difference of $\pm 20^\circ$ it was considered part of the same worm body, otherwise it was from another worm. The results after separating connected worms from Fig. 4a are shown in Fig. 4c, d, e.

In images of densely populated worms, the worms may be above or below each other, or part of their body may be wrapped around one or more other worms. For example, in Fig. 4 (see gray image Fig.

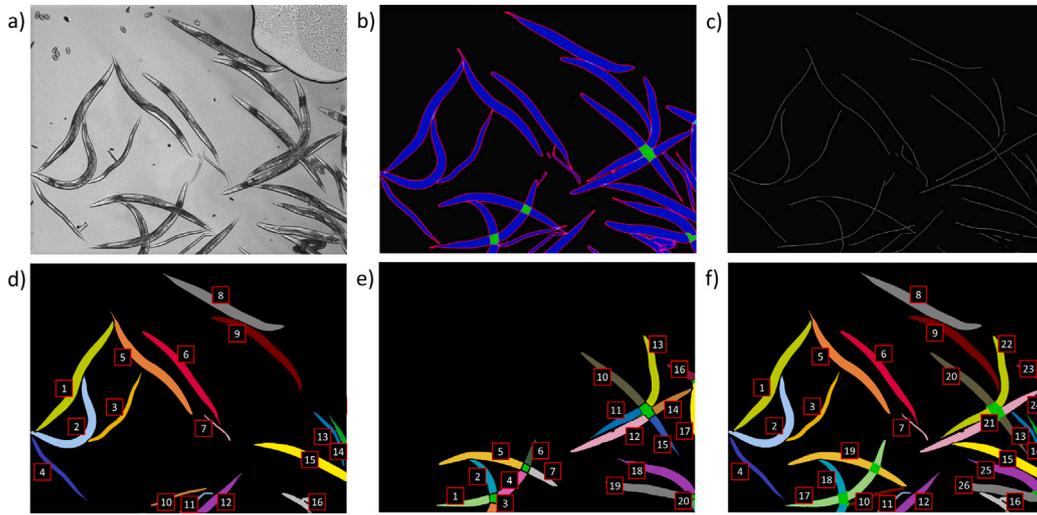


Fig. 2. Post-processing method. (a) Worms gray-image (input data to the proposed neural network model). (b) Worm segmentation into parts (worm edges, overlapping parts, and non-overlapping parts). (c) Segmentation of worm skeletons. (d) Non-overlapping worms (individual worms and worms in contact between edges or ends). (e) Overlapping worms divided into parts (each color represents a part of each worm and overlapping part in light-green). (f) Prediction of individual worms, each color represents each worm (non-overlapping worms [1–16] and separate overlapping worms [17–26]).

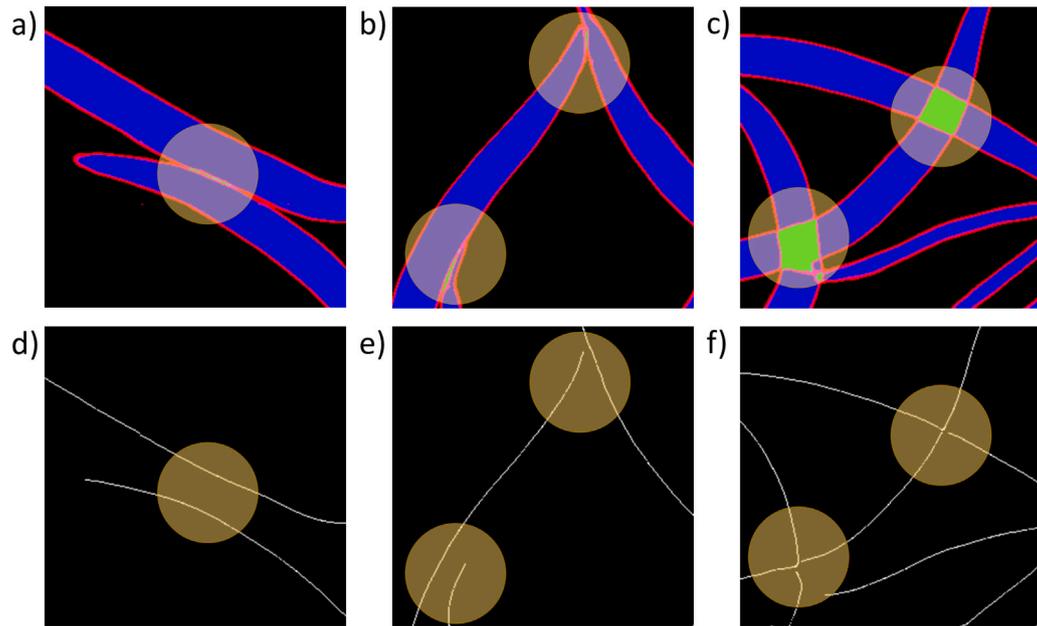


Fig. 3. Post-processing method, part 1. Identification of overlapping parts and parts in contact with other worms. (a) Overlapping worms at edges (edge contact). (b) Overlapping worms at head/tail ends (end contact). (c) Overlapping worms on other parts of the body (true-overlap). (d, e, f) Skeletons worms in overlapping parts, the yellow circle shows the overlapping area (light-green in Fig. 3 a, b, c).

2a), half of worm 1 (part 1, 4) is above worm 2 (part 2, 3), while the other half (part 6) is below worm 3 (part 5, 7). Although the worm is very flexible, in large populations it is very unlikely that parts of its body change abruptly. We rely on this assumption to connect all the separate parts. And no matter if one worm is above or below, the algorithm will connect the non-overlapping parts with the overlapping parts if the angle between non-overlapping parts meets the condition.

2.6. Results validation

The Intersection over Union (IoU) index was used to evaluate the accuracy of the results obtained from SegElegans. This metric measures the accuracy by dividing the intersection of areas or segmentations in our case by the union of these (Eq. (2)). The areas or segmentations used were the ground-truth (GT) and the results obtained after

applying the proposed neural network model and after the proposed post-processing method (PT).

$$IoU = \frac{GT \cap PT}{GT \cup PT} \tag{2}$$

To test the results of the post-processing with the ground-truth, two evaluations were carried out using the IoU metric. The first evaluated the result of the whole image, while the second evaluated the accuracy of the result per worm. The IoU value of the whole image result was obtained by integrating the separated worms from the ground-truth and the separated worms from the post-processing results into a single image respectively.

On the other hand, the value per worm was obtained after averaging the success values of each worm per image. For each ground-truth worm, its respective matching worm was found from post-processing. If

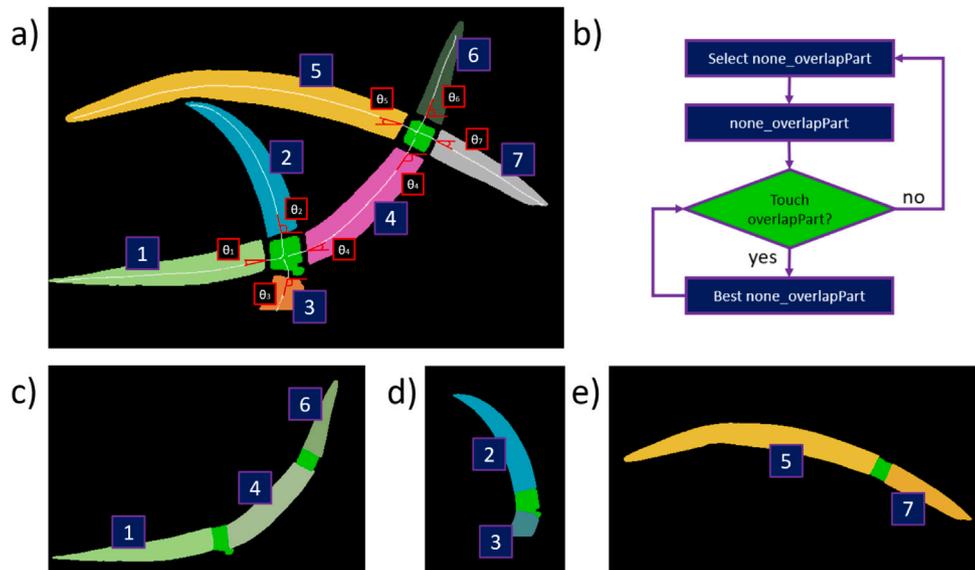


Fig. 4. Post-processing method, part 2. Separation of worms into true-overlap parts. (a) Split overlapping worms into none-overlapping parts [1–7] and true-overlapping parts (light-green). (b) Recursive function to join non-overlapping worm parts from overlapping worms. (c, d, e) Overlapping worm separation result.

the matching worm has an IoU value equal to or greater than 0.5, it was considered a success, i.e., a value of 1, in the event that no matching worm was found or the IoU value was less than 0.5, the value was zero.

The Matlab 2018b Machine Learning Toolbox software was used to compare the statistics between the proposed SegElegans method and EmbedSeg method. The p -value (0.05) from Kolmogorov–Smirnov Test (for large samples equal to or greater than 50 data) was used to check data normality. If the p -value was greater than or equal to 0.05, the null hypothesis H_0 was accepted (the data came from a normal distribution). Otherwise, the alternative hypothesis H_1 was accepted (the data did not come from a normal distribution). To assess the statistical significance of the results, Student's t -test was used if the data followed a normal distribution; otherwise, the Wilcoxon Signed Rank Test was employed.

3. Results

The ground-truth labels were obtained using the ImageJ software. A laboratory staff used this software to label each worm one by one in each of the training, validation, and test images. Manual tagging is done by creating a selection around the worm, typically using the polygon selection tool, in some implementation of the ImageJ image analysis software [4,5]. In our case we use the Fiji distribution, which comes prepackaged with a lot of relevant analysis plugins [3].

Training was performed with different windows to crop the input images, 256×256 , 512×512 (selected), and 768×768 . With 256×256 windows the output results of the networks had missing pixels in all segmentations especially in the edge and skeleton segmentations, while with the other window sizes the missing pixel errors were similar and minimal. Although the 768×768 window had a small improvement in results compared to the 512×512 window, the computational cost of using this window size was very high. The cost of processing an input image with a 768×768 window was 0.2 [ms] while with a 512×512 window it was 0.1 [ms], not to mention that with the 512×512 window, the training/validation/test were faster.

During training, dividing the input image into 64 windows of 512×512 pixels (8 horizontal and 8 vertical) allowed us to have a larger and more varied dataset, not to mention that the rotations and flips also helped to improve the generalization of the data and achieve the network convergence, avoiding overfitting.

Different original U-Net and modified U-Net architectures were trained and compared. The modifications consisted of changing each

double convolution block in the decoder of the original architectures for a convolution + LSTM block. The U-Net architectures used were: U-Net standard [20], Alexandre's U-Net [23], UMF U-Net [22], SmaAt DS [21] and SmaAT AT [21]. The results showed an average improvement using the modified U-Net architectures compared to the original ones, especially for images of segmentations worm parts Table 1.

The hardware used for training, validation, and testing was a X570 Pro4 machine, AMD Ryzen 9 3900X 12-Core Processor with 128 GB of RAM, and NVIDIA GeForce RTX 3090 graphics card, Ubuntu 20.04 64bits operating system. Our algorithm was implemented in a Python version 3.8.10 environment, using the Pytorch 1.8.1 libraries. The training and validation of the proposed neural network model took about 48 h with the hardware mentioned above. The hyper-parameters used for the training and validation phase were Batch_size = 6, num_workers = 6, maximum epoch = 500. The optimizer used was ADAM with a learning rate = 0.0001, betas = [0.95, 0.999], eps = $1e-8$, weight_decay = $1e-8$. CrossEntropyLoss() and BCEWithLogitsLoss() as loss functions for the training/validation step, and finally, the scheduler used was the lr_scheduler. LambdaLR with hyperparameter lr_lambda = lambda_fn, and last_epoch = epoch - 1. The function lambda_fn was Eq. (1).

During training the segmentation loss values of parts of worms and skeletons were obtained using the functions CrossEntropyLoss() and BCEWithLogitsLoss() respectively. If the average of the sum of both loss values (total loss) was less than the previous one, the neural network model was saved by overwriting the saved model. Table 1 shows the total parameters of each architecture as well as the average loss of the validation dataset and the average IoU of the test dataset for output1 (worm parts segmentation) and output2 (skeletons segmentation). All these results were obtained using the best neural model saved after training each model.

The results using one neural network model with one encoder and two decoders were very similar to using two independent models (one encoder and one decoder for each model respectively). In addition, using a single model instead of two allowed the training times as well as the computational cost in image processing to be reduced. On the other hand, replacing double convolution blocks in the decoder with ConvLSTM blocks helped improve the results of network output segmentations thanks to the advantage of LSTM blocks to obtain spatial, temporal, and dimensional features.

The training of all architectures (EmbedSeg, original U-Net, and modified U-Net) was performed using the same function and hyperparameters of the learning rate scheduler, the same training/validation

Table 1

Comparison of trained models. This table shows the results of the average loss and average intersection over union (IoU) metric for each output of the different neural network models evaluated.

Model	Parameters	Average loss		Average Iou	
		Segmentation	Skeleton	Segmentation	Skeleton
U-Net [20]	25.1111 M	0.1302	0.0178	0.1806	0.6718
U-Net enc + dec ConvLSTM	28.5775 M	0.1097	0.0181	0.7553	0.6781
U-Net A. [23]	25.1229 M	0.0937	0.0225	0.7691	0.6858
U-Net A. enc + dec ConvLSTM	28.5834 M	0.1142	0.0206	0.7847	0.6889
UMF U-Net [22]	25.1229 M	0.0855	0.0200	0.7635	0.6796
UMF U-Net enc + dec ConvLSTM	28.5834 M	0.1072	0.0193	0.7814	0.6800
SmaAt DS [21]	5.8391 M	0.0836	0.0155	0.7593	0.6790
SmaAt DS enc + dec ConvLSTM	21.3986 M	0.0981	0.0173	0.7761	0.6795
SmaAt AT [21]	25.2013 M	0.1018	0.0203	0.7700	0.6859
SmaAt AT enc + dec ConvLSTM	28.6618 M	0.1095	0.0186	0.7855	0.6907

Table 2

Comparison of architectures and models after post-processing. This table compares the final results obtained using the proposed method (SmaAt AT enc + Dec ConvLSTM) and post-processing method Section 2.5 with the method [26]. This table also shows a comparison with other U-Net architectures proposed by other authors.

	Avg. IoU [0.5]		Avg. IoU [0.7]		Avg. IoU [0.9]	
	Whole image	Per worm	Whole image	Per worm	Whole image	Per worm
EmbedSeg [26]	0.8775	0.9017	0.8230	0.8867	0.6891	0.7247
U-Net A. [23]	0.9287	0.9489	0.9287	0.9079	0.9287	0.9079
U-Net A. enc + dec ConvLSTM	0.9308	0.9531	0.9292	0.9265	0.9292	0.9265
UMF U-Net [22]	0.9266	0.9382	0.9266	0.9088	0.9266	0.9088
UMF U-Net enc + dec ConvLSTM	0.9335	0.9545	0.8607	0.9224	0.8607	0.9224
SmaAt DS [21]	0.9272	0.9238	0.9272	0.8895	0.9272	0.8895
SmaAt DS enc + dec ConvLSTM	0.9295	0.9610	0.9295	0.9311	0.9295	0.9311
SmaAt AT [21]	0.9343	0.9498	0.9343	0.9260	0.9343	0.9260
SmaAt AT enc + dec ConvLSTM	0.9355	0.9627	0.9355	0.9461	0.9335	0.9461

datasets as well as the data augmentation methods. After training, the test dataset was processed using the post-processing methods from EmbedSeg [26] and the method proposed in this work (Section 2.5) to obtain masks or segmentations of individual worms respectively. Table 2 shows the average results after using these post-processing methods for the whole image and per worm.

Figs. 5–7 show the average IoU values for all architectures with different threshold, as well as the mean (green line) and median (gray line) values using a box plot.

Statistical analysis was performed to compare the results obtained with (EmbedSeg [26]) and using the proposed method. 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 ($n \geq 50$). 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 3); therefore, the alternative hypothesis H1 was accepted, and the Wilcoxon Signed Range Test was used (Table 4, 5).

Finally, Fig. 8 shows a comparison between the results obtained using the proposed worm segmentation method (column three) and the manually labeled ground-truth images. Shown in the first column are the input images of the proposed neural network model. The result of each worm is shown with a color and a box with a number respectively. These results are saved in .tiff files, where each channel saves the mask or segmentation of each worm. A demo of the proposed method (proposed neural network method and post-processing method) is found in the source code section.

4. Discussion

The bioimaging community has flourished over the years thanks to the existence of ImageJ [33], which has provided a standardized platform that is reliable, openly available, highly versatile thanks to the myriad of community contributed plugins and accessible even to

Table 3

Normality test on the difference of methods (Proposed method – EmbedSeg). The p -value obtained was $1.87E-6$ 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.

	Diff
N	50
Minimum	-0.0170
Maximum	0.8674
Mean	0.0581
Std. Deviation	0.1599
p-value	1.872E-06

Table 4

Wilcoxon Signed Rank Test. The Wilcoxon signed-rank test table shows the difference between two related samples across positive, negative, and tied ranks.

	N	Mean rank	Sum ranks
Positive	30 ^a	31.6333	949
Negative	20 ^b	16.3000	326
Ties	0 ^c		
Total	50		

^a Proposed method > EmbedSeg

^b Proposed method < EmbedSeg

^c Proposed method = EmbedSeg

users with minimal technical understanding of image processing. The augmentation of ImageJ with the powers of machine/deep learning is the natural next step in the evolution of the platform. Several others have made their contribution to this evolution by developing new networks, ways to train networks inside ImageJ as well as methods to easily connect existing networks to the system [34–38]. SegElegans

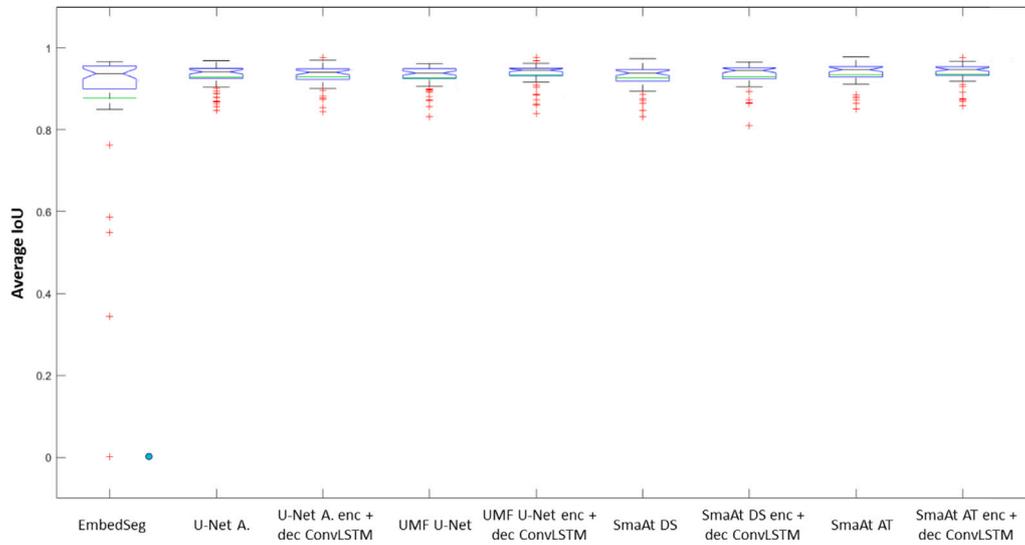


Fig. 5. Comparison of whole IoU Image average from all models using 0.5 threshold. EmbedSeg N = 50, mean = 0.8775, median = 0.9372, standard deviation = 0.1775, variance = 0.0315. U-Net A. N = 50, mean = 0.9287, median = 0.9416, standard deviation = 0.0315, variance = 0.0010. U-Net A. enc + dec ConvLSTM N = 50, mean = 0.9308, median = 0.9531, standard deviation = 0.0309, variance = 0.0010. UMF U-Net N = 50, mean = 0.9266, median = 0.9384, standard deviation = 0.0315, variance = 0.0010. UMF U-Net enc + dec ConvLSTM N = 50, mean = 0.9335, median = 0.9455, standard deviation = 0.0307, variance = 0.0009. SmaAt DS N = 50, mean = 0.9272, median = 0.9384, standard deviation = 0.0320, variance = 0.0010. SmaAt DS enc + dec ConvLSTM N = 50, mean = 0.9295, median = 0.9448, standard deviation = 0.0343, variance = 0.0012. SmaAt AT N = 50, mean = 0.9343, median = 0.9470, standard deviation = 0.0298, variance = 0.0009. SmaAt AT enc + dec ConvLSTM N = 50, mean = 0.9355, median = 0.9472, standard deviation = 0.0297, variance = 0.0009.

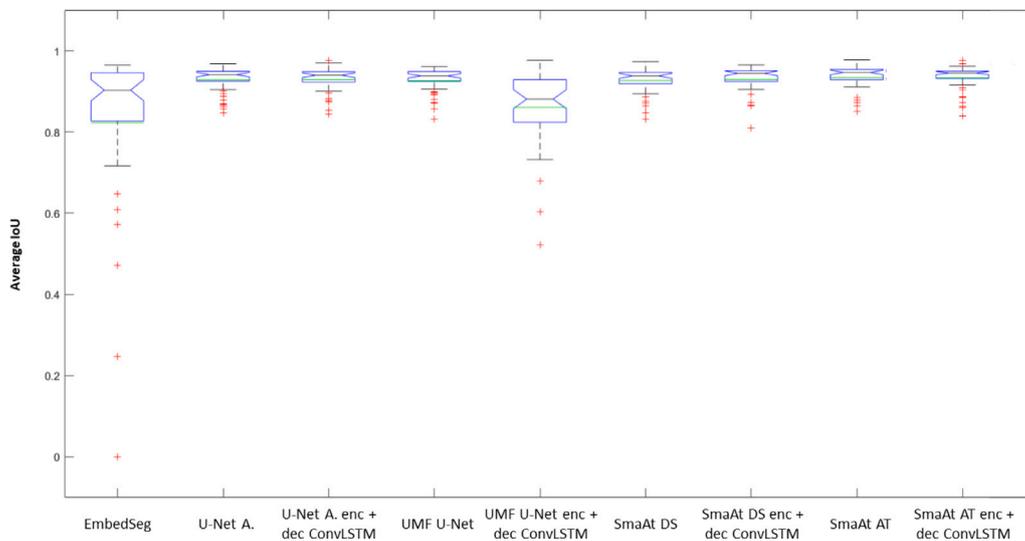


Fig. 6. Comparison of whole IoU Image average from all models using 0.7 threshold. EmbedSeg N = 50, mean = 0.8230, median = 0.9033, standard deviation = 0.2225, variance = 0.0495. U-Net A. N = 50, mean = 0.9287, median = 0.9416, standard deviation = 0.0315, variance = 0.0010. U-Net A. enc + dec ConvLSTM N = 50, mean = 0.9292, median = 0.9402, standard deviation = 0.0309, variance = 0.0010. UMF U-Net N = 50, mean = 0.9266, median = 0.9384, standard deviation = 0.0315, variance = 0.0010. UMF U-Net enc + dec ConvLSTM N = 50, mean = 0.8607, median = 0.8813, standard deviation = 0.0953, variance = 0.0091. SmaAt DS N = 50, mean = 0.9272, median = 0.9384, standard deviation = 0.0320, variance = 0.0010. SmaAt DS enc + dec ConvLSTM N = 50, mean = 0.9295, median = 0.9448, standard deviation = 0.0343, variance = 0.0012. SmaAt AT N = 50, mean = 0.9343, median = 0.9470, standard deviation = 0.0298, variance = 0.0009. SmaAt AT enc + dec ConvLSTM N = 50, mean = 0.9335, median = 0.9455, standard deviation = 0.0307, variance = 0.0009.

aims to be another contribution to this ecosystem, one trying to address an issue that is relative niche yet rather important to a specific part of the biomedical research community.

The analysis of microscopic images featuring large numbers of *C. elegans* (possibly dozens of individual animals) in ImageJ via manual segmentation can be a quite time-consuming and tedious process. Nevertheless it has been the preferable choice for worm researchers who sought to collect accurate data, since existing methods to obtain segmentations automatically have accuracy issues, especially in regards to dealing with the common phenomenon of touching or overlapping

worms. SegElegans resolves this issue and can simultaneously accurately differentiate individual worms while also obtaining excellent segmentation against the background and various types of noise (worm eggs, dark objects, debris, stains). It can even automatically detect (and suggest excluding) worms that may be unfit for analysis because they are damaged, obscured by the edge of the image or too small. It achieves all of this thanks to its novel dual architecture that allows us to extract the ideal information from multiple networks that are optimized for the detection of different features of the image. Our tests indicate that this architecture outperforms the next best recent instance

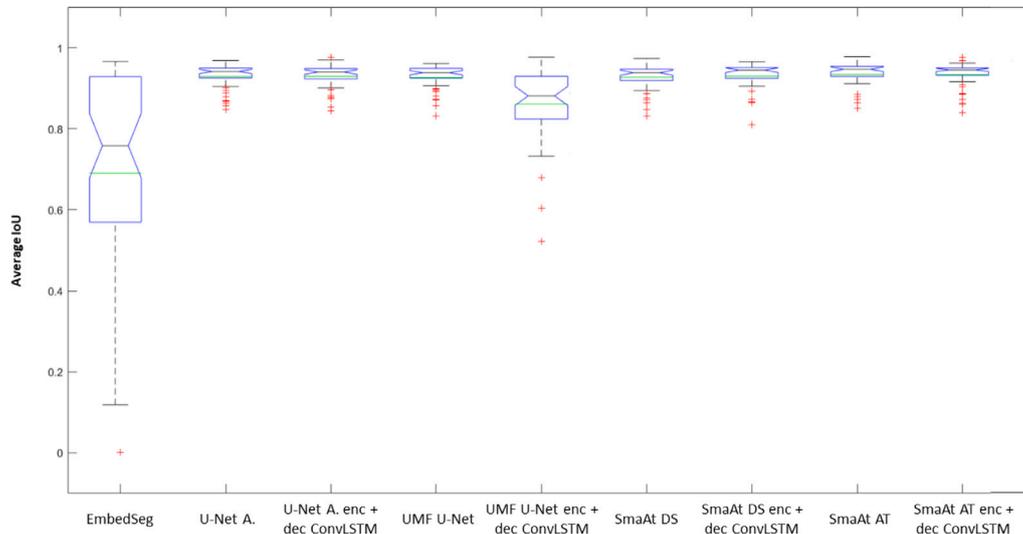


Fig. 7. Comparison of whole IoU Image average from all models using 0.9 threshold. EmbedSeg N = 50, mean = 0.6891, median = 0.7572, standard deviation = 0.2764, variance = 0.0764. U-Net A. N = 50, mean = 0.9287, median = 0.9416, standard deviation = 0.0315, variance = 0.0010. U-Net A. enc + dec ConvLSTM N = 50, mean = 0.9292, median = 0.9402, standard deviation = 0.0309, variance = 0.0010. UMF U-Net N = 50, mean = 0.9266, median = 0.9384, standard deviation = 0.0315, variance = 0.0010. UMF U-Net enc + dec ConvLSTM N = 50, mean = 0.8607, median = 0.8813, standard deviation = 0.0953, variance = 0.0091. SmaAt DS N = 50, mean = 0.9272, median = 0.9384, standard deviation = 0.0320, variance = 0.0010. SmaAt DS enc + dec ConvLSTM N = 50, mean = 0.9295, median = 0.9448, standard deviation = 0.0343, variance = 0.0012. SmaAt AT N = 50, mean = 0.9343, median = 0.9470, standard deviation = 0.0298, variance = 0.0009. SmaAt AT enc + dec ConvLSTM N = 50, mean = 0.9335, median = 0.9455, standard deviation = 0.0307, variance = 0.0009.

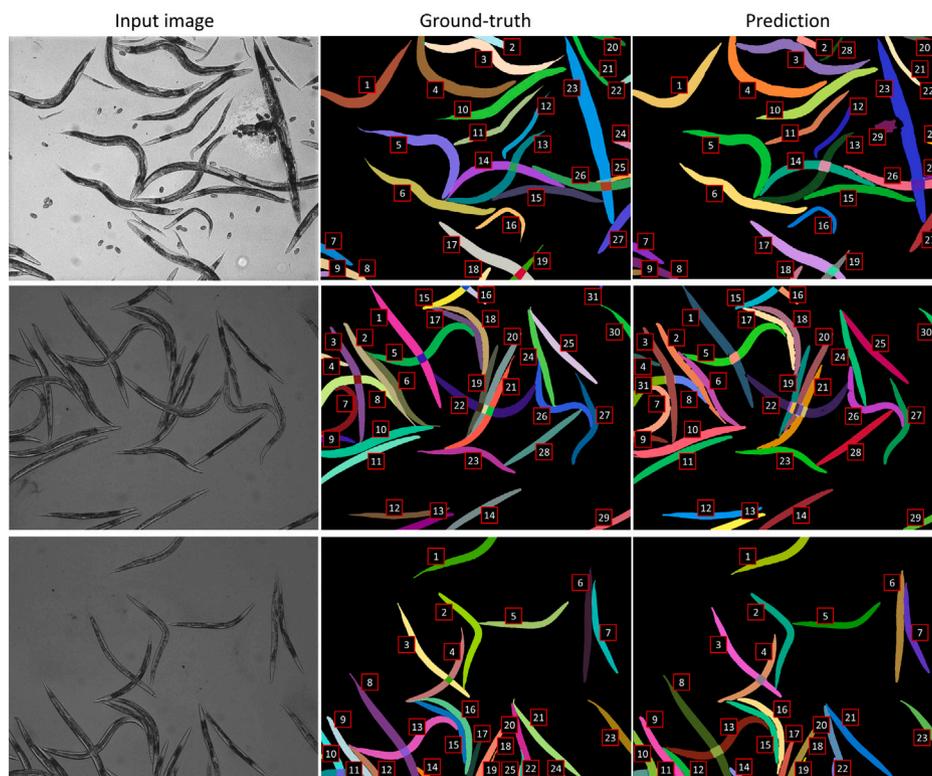


Fig. 8. Worm prediction results. The images on the left show the data input to the proposed neuronal network model, the center images show the ground-truth labels (GT), and the images on the right show the result of the proposed method. The IoU results for whole image are 0.9122, 0.8215, 0.8806, while the results per worm are 1, 0.9063, 0.96875 (row 1, 2, 3, respectively).

segmentation method (EmbedSeg) and various other neural network architectures even when using standard U-Net type network blocks. Further optimization using a SmaAT AT encoder block and ConvLSTM decoder blocks allows us to achieve an IoU score of 0.9627, ~5.8 better than EmbedSeg.

We are hopeful that SegElegans, even in this initial version, will be a great tool for *C. elegans* labs that will permit them to obtain results at a fraction of the time while also enjoying the consistency of an automatic method that is independent of user bias and proficiency. Following this first public release, we hope to be able to further improve the system's

Table 5

The p -value obtained with the Wilcoxon rank test was 0.0026 less than the significance value of 0.05, thus concluding there was a statistically significant difference between both models.

	Proposed method < EmbedSeg
z-val	-3.0071 ^b
p-value	0.002638

a. Wilcoxon Signed Ranks Test.

b. Based on negative ranks.

accuracy with larger scale training, optimize the postprocessing algorithm to be better and faster, train additional networks with the same architecture for the selection of individual *C. elegans* regions/body parts and provide additional ways to access SegElegans, ideally directly from within ImageJ [35].

5. Conclusions

The processing of numerous microscopic images, typically featuring 50–60 worms, can be challenging due to the tendency of worms to touch or cluster. The SegElegans method proposed in this paper allows obtaining individual segmentations of *C. elegans* in microscopic images. SegElegans effectively separates overlapping *C. elegans* and provides excellent background segmentation, removing various types of noise (worm eggs, dark objects, debris, stains). (See Fig. 7 for details) We also meticulously assessed the efficacy of our novel dual architecture approach. Interestingly, SegElegans outperforms the best recent instance segmentation method (EmbedSeg) and various neural network architectures. The results showed better average IoU values with U-Net type architectures than with a recent instance segmentation method (EmbedSeg). The best results were obtained with SegElegans method utilizing the SmaAtATenc+decConvLSTM neural networks which had an improvement of 5.8% over the EmbedSeg method. To enhance the accuracy of individual worm predictions, future research should prioritize expanding dataset availability, refining hyperparameter adjustments, exploring advanced segmentation algorithms, and enhancing post-processing techniques.

SegElegans is an effective application for researchers, aiding in the segmentation of entire worms. Importantly, the produced whole-body segmentations/masks from brightfield images can be co-registered with darkfield fluorescent images (obtained from the same imaging system) to facilitate more sophisticated analyses such as marker quantification in microscopic images.

6. Source code

The proposed method was developed in Ubuntu-Linux 20.04 64 bits using python3.8.10 with the Pytorch 1.8.1 libraries. The source code is on GitHub; it is open source and can be downloaded from the repository at <https://github.com/playanaC/SegElegan.git>. The demo application can be used at the googlecolab link:

https://github.com/playanaC/SegElegan/blob/main/1_Demo_predict.ipynb.

CRediT authorship contribution statement

Pablo E. Layana Castro: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Konstantinos Kounakis:** Writing – review & editing, Validation, Conceptualization. **Antonio García Garvía:** Writing – review & editing, Validation, Software, Conceptualization. **Ilias Gkikas:** Writing – review & editing, Validation. **Ioannis Tsiamantas:** Writing – review & editing, Validation. **Nektarios Tavernarakis:** Writing – review & editing, Supervision, Resources, Investigation, Formal analysis, Conceptualization. **Antonio-José Sánchez-Salmerón:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare no competing interests.

Acknowledgments

We thank Aggela Pasparki for the technical assistance and all members of our labs for the useful discussions. This study was supported by Universidad Politécnica de Valencia, Spain through Instituto de Automática e Informática Industrial, FPI Predoctoral contract PRE2019-088214, Ministerio de Universidades (Spain) under grant FPU20/02639 and by European FEDER funds. The authors also thank the EU-FEDER Comunitat Valenciana 2014–2020 grant IDIFEDER/2018/025 and European Union Horizon 2020 FETOPEN, project “Dynamic”, under the grant agreement “GA-863203”, the European Research Council, under grant agreement “ERC-GA695190-MANNA” to N.T., the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), Greece, under grant agreement No [1898] and No [2040] to E.L.

References

- [1] D. Biron, G. Haspel (Eds.), *C. elegans*, in: Methods and Applications, Springer Science+Business Media, New York, 2015, p. 12, <http://dx.doi.org/10.1007/978-1-4939-2842-2>.
- [2] A. Olsen, M.S. Gill (Eds.), *Ageing: Lessons from C. elegans*, Springer International Publishing, Switzerland, 2017, <http://dx.doi.org/10.1007/978-3-319-44703-2>.
- [3] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al., Fiji: an open-source platform for biological-image analysis, *Nature Methods* 9 (7) (2012) 676–682, <http://dx.doi.org/10.1038/nmeth.2019>.
- [4] J. Schindelin, C.T. Rueden, M.C. Hiner, K.W. Eliceiri, The ImageJ ecosystem: An open platform for biomedical image analysis, *Mol. Reprod. Dev.* 82 (7–8) (2015) 518–529, <http://dx.doi.org/10.1002/mrd.22489>.
- [5] C.T. Rueden, J. Schindelin, M.C. Hiner, B.E. DeZonia, A.E. Walter, E.T. Arena, K.W. Eliceiri, ImageJ2: Imagej for the next generation of scientific image data, *BMC Bioinformatics* 18 (2017) 1–26, <http://dx.doi.org/10.1186/s12859-017-1934-z>.
- [6] K. Ramki, G. Thirupathi, S.K. Ramasamy, P. Sundararaj, P. Sakthivel, An aggregation-induced emission-based ratiometric fluorescent chemosensor for hg(II) and its application in *Caenorhabditis elegans* imaging, *Methods* 221 (2024) 1–11, <http://dx.doi.org/10.1016/j.ymeth.2023.11.010>.
- [7] H. Wang, U. Karadge, W.H. Humphries, A.L. Fisher, Analyzing cell physiology in *C. elegans* with fluorescent ratiometric reporters, *Methods* 68 (3) (2014) 508–517, <http://dx.doi.org/10.1016/j.ymeth.2014.05.012>, *C. elegans Methods*.
- [8] Y. Xiao, L. Zhang, Y. Liu, Protocol for assessing the healthspan of *Caenorhabditis elegans* after potential anti-aging drug treatment, *STAR Protoc.* 4 (2) (2023) 102285, <http://dx.doi.org/10.1016/j.xpro.2023.102285>.
- [9] M. Akdag, V. van Schijndel, T. Sinnige, Islet amyloid polypeptide tagged with green fluorescent protein localises to mitochondria and forms filamentous aggregates in *Caenorhabditis elegans*, *Biophys. Chem.* 307 (2024) 107180, <http://dx.doi.org/10.1016/j.bpc.2024.107180>.
- [10] K.W. Oh, D.-K. Kim, A.-L. Hsu, S.-J. Lee, Distinct sets of lysosomal genes define synucleinopathy and tauopathy, *BMB Rep.* 56 (12) (2023) 657.
- [11] N.B. Rizvandi, A. Pižurica, F. Rooms, W. Philips, Skeleton analysis of population images for detection of isolated and overlapped nematode *C. elegans*, in: 2008 16th European Signal Processing Conference, IEEE, 2008, pp. 1–5.
- [12] N.B. Rizvandi, A. Pizurica, W. Philips, Machine vision detection of isolated and overlapped nematode worms using skeleton analysis, in: 2008 15th IEEE International Conference on Image Processing, IEEE, 2008, pp. 2972–2975, <http://dx.doi.org/10.1109/ICIP.2008.4712419>.
- [13] C. Wählby, L. Kamensky, Z.H. Liu, T. Riklin-Raviv, A.L. Conery, E.J. O’rourke, K.L. Sokolnicki, O. Visvikis, V. Ljosa, J.E. Irazoqui, et al., An image analysis toolbox for high-throughput *C. elegans* assays, *Nature Methods* 9 (7) (2012) 714–716, <http://dx.doi.org/10.1038/nmeth.1984>.
- [14] V. Uhlmann, M. Unser, Tip-seeking active contours for bioimage segmentation, in: 2015 IEEE 12th International Symposium on Biomedical Imaging, ISBI, 2015, pp. 544–547, <http://dx.doi.org/10.1109/ISBI.2015.7163931>.
- [15] L. Mais, P. Hirsch, D. Kainmueller, PatchPerPix for instance segmentation, in: European Conference on Computer Vision, vol. 12370, Springer, Cham, 2020, pp. 288–304, http://dx.doi.org/10.1007/978-3-030-58595-2_18.
- [16] J.L. Rumberger, L. Mais, D. Kainmueller, Probabilistic deep learning for instance segmentation, in: European Conference on Computer Vision, Springer, 2020, pp. 445–457, http://dx.doi.org/10.1007/978-3-030-66415-2_29.

- [17] S. Wiehman, H. de Villiers, Semantic segmentation of bioimages using convolutional neural networks, in: 2016 International Joint Conference on Neural Networks, IJCNN, IEEE, Vancouver, BC, Canada, 2016, pp. 624–631, <http://dx.doi.org/10.1109/IJCNN.2016.7727258>.
- [18] L. Chen, M. Strauch, M. Daub, X. Jiang, M. Jansen, H.-G. Luigs, S. Schultzkühlmann, S. Krüssel, 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, IEEE, Iowa City, IA, USA, 2020, pp. 508–512, <http://dx.doi.org/10.1109/ISBI45749.2020.9098465>.
- [19] S. Fudickar, E.J. Nustede, E. Dreyer, J. Bornhorst, Mask R-CNN based C. *Elegans* detection with a DIY microscope, Biosensors 11 (8) (2021) <http://dx.doi.org/10.3390/bios11080257>.
- [20] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 9351, Springer, Cham, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.
- [21] K. Trebing, T. Stanczyk, S. Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-unet architecture, Pattern Recognit. Lett. 145 (2021) 178–186, <http://dx.doi.org/10.1016/j.patrec.2021.01.036>.
- [22] E. Plebani, N.P. Biscola, L.A. Havton, B. Rajwa, A.S. Shemonti, D. Jaffey, T. Powley, J.R. Keast, K.-H. Lu, M.M. Dunder, High-throughput segmentation of unmyelinated axons by deep learning, Sci. Rep. 12 (1) (2022) 1–16, <http://dx.doi.org/10.1038/s41598-022-04854-3>.
- [23] M. Alexandre, Pytorch-UNet, 2019, Code <https://github.com/milesial/Pytorch-UNet>.
- [24] P.E. Layana Castro, A.G. Garvía, A.-J. Sánchez-Salmerón, Automatic segmentation of *Caenorhabditis Elegans* skeletons in worm aggregations using improved U-Net in low-resolution image sequences, Heliyon 9 (4) (2023) e14715, <http://dx.doi.org/10.1016/j.heliyon.2023.e14715>.
- [25] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969, <http://dx.doi.org/10.1109/TPAMI.2018.2844175>.
- [26] M. Lalit, P. Tomancak, F. Jug, EmbedSeg: Embedding-based instance segmentation for biomedical microscopy data, Med. Image Anal. 81 (2022) 102523, <http://dx.doi.org/10.1016/j.media.2022.102523>.
- [27] F.I. Diakogiannis, F. Waldner, P. Caccetta, C. Wu, ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data, ISPRS J. Photogramm. Remote Sens. 162 (2020) 94–114, <http://dx.doi.org/10.1109/ACCESS.2021.3111131>.
- [28] N. Micallef, D. Seychell, C.J. Bajada, Exploring the u-net++ model for automatic brain tumor segmentation, Ieee Access 9 (2021) 125523–125539, <http://dx.doi.org/10.1109/ACCESS.2021.3111131>.
- [29] M.Z. Alom, C. Yakopcic, T.M. Taha, V.K. Asari, Nuclei segmentation with recurrent residual convolutional neural networks based U-net (R2U-net), in: NAECON 2018-IEEE National Aerospace and Electronics Conference, IEEE, 2018, pp. 228–233, <http://dx.doi.org/10.1109/NAECON.2018.8556686>.
- [30] T. Stiernagle, Maintenance of *C. Elegans*, in: The *C. Elegans* Research Community, WormBook, 2006, pp. 2005–2018, <http://dx.doi.org/10.1895/wormbook.1.101.1>, URL <https://www.ncbi.nlm.nih.gov/books/NBK19649/?report=classic>.
- [31] DavideA, ConvLSTM_pytorch, 2020, Code https://github.com/ndrplz/ConvLSTM_pytorch.
- [32] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, Adv. Neural Inf. Process. Syst. 28 (2015).
- [33] A.B. Schroeder, E.T. Dobson, C.T. Rueden, P. Tomancak, F. Jug, K.W. Eliceiri, The ImageJ ecosystem: Open-source software for image visualization, processing, and analysis, Prot. Sci. 30 (1) (2021) 234–249, <http://dx.doi.org/10.1002/pro.3993>.
- [34] I. Arganda-Carreras, V. Kaynig, J. Schindelin, A. Cardona, H. Seung, Trainable weka segmentation: A machine learning tool for microscopy image segmentation, neurosci. 2014 short course 2-adv. Brain-scale, autom, Anat. Tech. Neuronal Reconstr. Tract Tracing, Atlasing (2014) 73–80, <http://dx.doi.org/10.1093/bioinformatics/btx180>.
- [35] E. Gómez-de Mariscal, C. García-López-de Haro, W. Ouyang, L. Donati, E. Lundberg, M. Unser, A. Muñoz-Barrutia, D. Sage, DeepImageJ: A user-friendly environment to run deep learning models in ImageJ, Nature Methods 18 (10) (2021) 1192–1195, <http://dx.doi.org/10.1038/s41592-021-01262-9>.
- [36] S.J. Yang, M. Berndl, D. Michael Ando, M. Barch, A. Narayanaswamy, E. Christiansen, S. Hoyer, C. Roat, J. Hung, C.T. Rueden, et al., Assessing microscope image focus quality with deep learning, BMC Bioinformatics 19 (2018) 1–9, <http://dx.doi.org/10.1186/s12859-018-2087-4>.
- [37] T.-A. Song, F. Yang, J. Dutta, Noise2Void: unsupervised denoising of PET images, Phys. Med. Biol. 66 (21) (2021) 214002, <http://dx.doi.org/10.1088/1361-6560/ac30a0>.
- [38] T.-O. Buchholz, M. Prakash, D. Schmidt, A. Krull, F. Jug, DenoiSeg: joint denoising and segmentation, in: European Conference on Computer Vision, Springer, 2020, pp. 324–337, http://dx.doi.org/10.1007/978-3-030-66415-2_21.